DATABASE ARCHITECTURE AND QUERY STRUCTURES FOR PROBE DATA PROCESSING

by

Jason S. Wasson
Indiana Department of Transportation
P.O. Box 2279
West Lafayette, IN 47906
Phone: (765) 463-1521
Fax: (765) 497-1665
Email: jwasson@indot.in.gov

Corresponding author:
Darcy M. Bullock
Purdue University
550 Stadium Mall Dr
West Lafayette, IN 47906
Phone: (765) 496-7314
Fax: (765) 496-7996
Email: darcy@purdue.edu

January 12, 2012

ABSTRACT

Probe data can be used to characterize the movement of people and vehicles within and between various modes of transportation. Regardless of the technology used to obtain the unique or pseudo-unique identifiers, all probe data reduction techniques require a systematic method for matching identifiers observed at different points in the transportation system. Although algorithms for matching of unique identifiers observed at different locations have been developed in an ad-hoc manner for many years, the magnitude of unique identifiers that can be collected from electronic devices is much larger. Although the literature is rich on probe vehicle data collection technologies and statistical processing techniques, the literature is silent on scalable database architectures that can be used to manage large probe vehicle data sets in a systematic manner. This paper frames the probe data management problem and describes a generalized framework for the use of a relational database to systematically match unique probe vehicle identifiers.

MOTIVATION

Probe data can be used to characterize the movement of people and vehicles within and between various modes of transportation. Probe data processing relies on matching observations of unique identifiers associated with a traveler, such as a license plate, at two or more locations on a network. While the use of probe data has existed for decades, transitioning from matching license plates to matching electronic devise such as toll tags and consumer electronic devices has vastly increased the scalability and cost effectiveness of collecting probe data within and across various transportation modes [1-14].

Regardless of the technology used to obtain the unique or pseudo-unique identifiers, all probe data reduction techniques require a systematic method for matching identifiers observed at different points in the transportation system. Although algorithms for matching of unique identifiers observed at different locations have been developed in an ad-hock manner for many years, the magnitude of unique identifiers that can be collected from electronic devices is much larger. Although algorithmic approaches work quite well, they are frequently encapsulated as a black-box module. A more scalable approach is to leverage the searching and matching capabilities of modern relational database engines to provide a suite of queries to derive information such as segment space-mean speed, segment travel times, origin-destination pairs, route choice characteristics, and modal choice.

The large numbers of possible match permutations necessities a good database architecture and matching algorithm to facilitate efficient processing of the data. This paper presents a framework for the use of a relational database in the context of SQL based data reduction strategies using probe identifiers collected from Bluetooth® enabled devices observed along roadways in Indiana. These analytical methods are technology independent and transferable to other probe based technologies across transportation system modes.

PROBE IDENTIFIERS

The fundamental data elements of probe datasets consists of the probe’s unique identifier, location, and associated time it was observed at a known location in the network. Analysis of these fundamental data elements can be performed using structured data queries comparing observation times at one or more locations for a given matched probe identifier. Structured Query Language (SQL) is a mature and standardized language used to manage data in relational database management systems (RDBMS), which are widely used in the storage of transportation data by transportation agencies, consultants, and university institutions.

Figure 1a-c represents a probe vehicle traveling along a signalized corridor with multiple mid-block probe data acquisition stations. The circular markers shown in Figure 1 illustrate the progression of the probe vehicle per uniform time interval. This figure illustrates that multiple successful inquiry communications sessions can exist between the mobile probe vehicle and the stationary roadside collection sites on a single pass of the vehicle through the system. During the stationary dwell period and velocity transition periods, time elapses and multiple inquiry communication sessions occur between the probe vehicle and the roadside equipment. Figure 1b shows an enlarged link for Figure 1a in which the probe vehicle exhibits a variance in speed due to the queuing and subsequent discharge between intersections 1 and 2. The number of these logged communication transactions largely depends on the probe vehicle speed, and the effective size of the detection zone, which is a factor of the antenna characteristics and the RF power output, polling interval, and sensitivity of the radio transceivers. This repetitive acquisition of communication inquiry response messages from a moving or stationary probe vehicle results in multiple appearances of the associated vehicle identifier in the data stream for any particular station location. These duplicative probe identifiers are not isolated to signalized arterials as this same phenomenon can be seen from other queuing scenarios such as freeway congestion, toll booths or security screening checkpoints. Figure 1c represents a time space diagram for the same vehicle traversing the roadway segment between intersections I_1 and I_2 shown in Figure 1b.

Due to the multiple logging of a probe vehicle's unique identifier (a MAC address in this example) as it passes a roadside data collection station, there exist multiple matching options when computing travel time between two separated data collection stations as shown in Figure 2. The first-first matching methodology is perhaps the most prevalent in the literature when conducting travel time evaluations [1–3], [5–7], [10–12], but some studies' design of experiment may benefit from the alternate matching sequences shown. Other possible matching algorithms could include: first-last, last-first, last-last or one based on maximum signal strength. While the ideal-ideal matching option shown in Figure 2 in which the wireless probe observation corresponds to the probe occupying the representative center of a detection zone, it is often not practical to implement. Therefore, some temporal and spatial errors are inherent in the acquired data based on probes being in different locations within the detection zone.

Travel patterns of motorists driving vehicles emitting an electronic identifier along a roadway corridor can result in the repeat appearance of that probe's identifier over time due to different trips within the network. Developing techniques to properly segment these identifiers into trips is important for an algorithm or query that processes probe data. Figure 3 shows the appearance over an 14-hour period of three unique vehicle identifiers at two detection stations, station 1 in Figure 3a and station 2 in Figure 3b, which are separated by over 5 miles. The first vehicle identifier of 00:00:A0:39:13:CD shows a single trip between station 1 and station 2 near 0800 hours. The second vehicle identifier of 00:00:A0:4E:0F:F2 shows a single trip between station 1 and station 2 near 1100 hours as well as a second appearance at station 2 near 1815 hours. The third vehicle identifier of 00:00:A0:62:CE:A7 shows two trips between station 1 and station 2 at approximately 0900 and 1315 hours respectively as well as two other observations at station 2 near 1200 and 1500 hours. The reappearance of the unique probe identifier in the observed dataset over time necessitates the consideration of time of day computational windows when searching for identifier matches between two locations.

An array of probe data collection stations within a transportation system network, represented by the variable n, the number of matching permutations between all origin-destination (o-d) pairs escalates based on the equation of $n(n-1)$. This non-linear escalation of o-d pairs underscores the need for an efficient and scalable data reduction methodology.

DATA REDUCTION METHODOLOGY

This section will describe a data reduction methodology using standard SQL syntax which will compute travel times of unique probe identifiers between two roadside data acquisition stations. All queries shown in accompanying figures were developed under PostgreSQL version 9.1 [13]. Three strategies will be presented from simple to complex that refines the data reduction methodology to account for multiple appearances of probe identifiers in the data stream.

The strategies described are presented as a series of cascading views which use the results from the previous step to further refine the data reduction process. An overview illustration of the views' relationships for each strategy will be shown as an entity-relationship. In a database management system, a view is simply a stored query that produces a "virtual" table which can be referenced like any other physical base table. The use of views is considered a good practice in the design of databases and associated applications and provides a convenient abstraction tool for decomposing complex queries into a series of simple queries that are easier to explain and document. [14] It should be noted that in the data reduction strategy presented by the series of cascading views can be replaced with an equivalent single SQL statement. The authors' use of the cascading view approach is to break down the data reduction strategy into discernible steps and should not imply any advantage over a single SQL statement which can often achieve greater efficiencies within the ORDBMS environment.

Upstream and downstream stations are referenced through the strategies discussed and imply directionality of the prove vehicle trajectory. Figure 1a shows the probe trajectory progressing from the upstream most station of S1 to the most downstream station of S6. For the purpose of the data reduction strategies presented S1 is considered the upstream station whereas S2 is that of the downstream station.

Strategy A: Matching vehicle identifiers between upstream and downstream stations

As previously described, the fundamental data record for probe data consists of a probe identifier, time of observation, and acquisition site identifier as shown in the data table illustrated in Figure 4a. The first stepwise view for Strategy A defined by SQL in Figure 4b consists of filtering all the data records in the data table to produce a subset of the data for the specified upstream and downstream stations for the time period of interest. In the example shown, all records in the data table are filtered to create a view, view_a01, which shows only the observations at station s1 and s2 that occur between 0600 and 2000 hours on July 13th, 2011. Table 1 shows the truncated results from view_a01. Figure 4c and Figure 4d simply filter the results view_a01 to segregate by the upstream and downstream stations of s1 and s2 as view_a02_upstream and view_a02_downstream respectively. The computation of travel time is made in Figure 4e which produces view_a03 in which travel time is presented in seconds. The 'date_part' SQL function converts the time of observation at station 1 and 2 to seconds using the 'epoch' modifier which references time as the offset in seconds from January 1, 1970. The SQL statement as presented requires that the downstream time stamp occurs after the upstream time stamp. This requirement results in the filtering of matches that occur in the opposite travel direction. For studies that need two-way trajectories for space-mean calculations this requirement can be omitted. Such two-way studies will have the directions designated by positive and negative values. Truncated results from view_a03 of Strategy A are shown in Table 2. Examination of Table 2 illustrates an issue where unique identifiers that are observed multiple times appear more than once in the computation of travel times. Subsequent strategies develop more sophisticated queries to address this issue.

Strategy B: Strategy A plus defined time interval bins and SQL aggregate functions

Strategy A above resulted in every possible upstream and downstream observation permutation being calculated as a travel time record. Therefore, additional SQL logic must be introduced to properly constrain the output. Strategy B introduces the use of SQL aggregate functions. For this paper, the first-first matching option was chosen to address multiple appearances of a vehicle identifier which a defined time horizon at a station. Definition of a time horizon is important to allow vehicle identifiers to be reused over time as global application of the SQL aggregate functions would allow an identifier to be reported only once in the resultant view. Strategy B imposes a time horizon, binning interval, of 1 hour or 3,600 seconds. Figure 5c and Figure 5d apply the minimum SQL aggregate function over a time binning interval of 3,600 seconds for the upstream and downstream stations respectively. This results in only the earliest occurrence of a vehicle identifier being used each hour regardless of the number of logged observations. The selection of a 3,600 second binning interval was subjective based on engineering judgment and should be carefully considered depending on the environment being evaluated. For example, a corridor with recurring trips throughout a day would suffer from a 24-hour binning interval in which a vehicle identifier could only be used once per day. Figure 5e shows the SQL statement which utilizes the filtered station observations for the computation of travel time. Note that the SQL aggregate function is also used in this travel time calculation step to ensure the upstream timestamp is used with the closest downstream timestamp for a specific vehicle identifier within a binning interval.

Comparison of results between Table 2 from Strategy A and Table 3 from Strategy B show the results of applying reasonable constrains in the data reduction process. For example, Table 2 for lines 14-21 in Strategy A show multiple travel time computational results for vehicle identifier 00:00:A0:62:CE:A7 based on two (2) observations at upstream station s1 at 09:00:26 and 13:24:14 respectively. If a vehicle passed this upstream station twice, it would be expected that only two corresponding travel time values be presented as opposed to the eight (8) listed in Table 2. Conversely, Table 3 lines 6-7 from Strategy B show only two (2) travel time computational results for the vehicle identifier of 00:00:A0:62:CE:A7 as one would expect for two possible trips along the corridor.

Strategy C: Strategy B plus vehicle identifier offset constraints between adjacent bin intervals

Strategy B provided significant refinement of the travel time calculation through the introduction of aggregate functions applied within time bin intervals. In the dataset used for this paper, the corresponding record count produced in Strategy B was reduced by 86% (from 12,907 to 1,770) when compared to the unconstrained approach in Strategy A. However, the time binning interval, introduced in Strategy B, established a time horizon in which a unique vehicle identifier can only occur once within a specified period of time. This process could result in the same vehicle identifier appearing in adjacent time interval bins. For example, Table 3 lines 3-4 for vehicle identifier 00:00:A0:4E:0F:F2 are based on two observations at upstream station 1 at 10:59:59 and 11:00:03. While these observations are only four (4) seconds apart and likely the result of a single pass of the probe vehicle through the detection zone, both results are displayed due to them occurring in adjacent hourly time bin intervals. Since the application of the time bin interval simply slices elapsed time of day into unit blocks of fixed width, it is possible to have vehicle identifiers, which are separated in time by less than bin interval, appear within adjacent time bins. The purpose of Strategy C is to further refine the results of Strategy B by filtering the appearances of a vehicle identifier that appears within adjacent time bin intervals but have a time offset less than the associated bin size.

Figure 7a,b evaluates through SQL logic each earliest appearance of a vehicle identifier within a time bin interval to determine if it also appeared within the previous time bin with a time of day offset less than the bin size. Table 4 shows the truncated results from the application of SQL shown in Figure 7c. Whereas Table 3lines 3-4 showed two results for vehicle identifier 00:00:A0:4E:0F:F2, application of the enhanced SQL in Strategy C result in this identifier appearing only once in Table 4 line 3. The additional data reduction impact of Strategy C reduced the number of resultant records by 21 records to 1,749. This corresponds to a 0.002% reduction over that of Strategy A and 0.012 % reduction over that of Strategy B. The impact of the SQL enhancements used in Strategy C will vary based on the relative size of the binning interval as well as the extent of congestion adjacent to probe data collection sites. Shorter time bin intervals with more congested conditions will result in more observation overlaps between adjacent time interval bins.

CONCLUSION

Matching unique probe identifies has historically been undertaken simplistically in an ad-hoc manner using spreadsheet or through vendors developing proprietary algorithmic approaches. This paper identified the matching of probe data unique identifiers as a fundamental probe data processing task that require algorithms that specifically document there approach to handling the matching operations and spatial uncertainty associated with multiple samples.

The data processing task associated with the multiple acquisitions of unique probe identifiers was identified and addressed through a flexible technique using a RDBMS approach that is both scalable and transparent in regards to its data reduction techniques. The associated problem of over filtering of acceptable reoccurrence of unique probe identifier through multiple trips within a transportation system was also identified and addressed through application of a systematic temporal discrete binning technique.

The platform independent SQL techniques presented in Figure 6 and Figure 7 provide a flexible, scalable and systematic approach for parameterized queries that provide efficient ad-hoc analysis of probe data for very large datasets. These queries are used on a daily basis by the Indiana DOT for processing 6 million probe vehicle identifiers every month from a network of nearly 30 locations.

ACKNOWLEDGMENTS

This work was supported by the Joint Transportation Research Program administered by the Indiana Department of Transportation and Purdue University. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official views or policies of the National Academy of Sciences, American Association of State Highway and Transportation Officials, Federal Highway Administration or the Indiana Department of Transportation sponsoring organizations. These contents do not constitute a standard, specification, or regulation.

REFERENCES

[1] J. S. Wasson, J. R. Sturdevant, and D. M. Bullock, “Real-Time Travel Time Estimates Using Media Access Control Address Matching,” *ITE Journal*, vol. 78, no. 6, pp. 20-23, Jun. 2008.

[2] S. M. Quayle, P. Koonce, D. DePencier, and D. M. Bullock, “Arterial Performance Measures with Media Access Control Readers,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2192, pp. 185-193, Dec. 2010.

[3] A. Haghani, M. Hamedi, K. Sadabadi, S. Young, and P. Tarnoff, “Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2160, pp. pp 60-68, 2010.

[4] T. M. Brennan, J. M. Ernst, C. M. Day, D. M. Bullock, J. V. Krogmeier, and M. Martchouk, “Influence of Vertical Sensor Placement on Data Collection Efficiency from Bluetooth MAC Address Collection Devices,” *Journal of Transportation Engineering*, vol. 136, p. 1104, 2010.

[5] R. J. Haseman, J. S. Wasson, and D. M. Bullock, “Real-Time Measurement of Travel Time Delay in Work Zones and Evaluation Metrics Using Bluetooth Probe Tracking,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2169, pp. 40-53, Dec. 2010.

[6] J. Porter, D. Kim, and M. Magana, “Wireless Data Collection System for Real-Time Arterial Travel Time Estimates,” Oregon Transportation Research and Education Consortium, Final Report OTREC-RR-10-16, Mar. 2011.

[7] D. Puckett and M. Vickich, “Bluetooth®-Based Travel Time/Speed Measuring Systems Development,” Texas Transportation Institute, Final Report UTCM 09-00-17, Jun. 2011.

[8] “Specification of the Bluetooth System, Part B: Baseband Specification.” The Bluetooth Special Interest Group, 01-Dec-1999.

[9] “IEEE Standard 802-2001.” The Institute of Electrical and Electronics Engineers, Inc. (IEEE), 07-Feb-2002.

[10] J. Wasson et al., “Evaluation of Spatial and Temporal Speed Limit Compliance in Highway Work Zones,” *Transportation Research Record*, vol. Publication Pending, p. 28p, 2011.

[11] Y. Malinovskiy, Y.-J. Wu, Y. Wang, and U. K. Lee, “Field Experiments on Bluetooth-Based Travel Time Data Collection,” in *Proceedings of the 89th Annual Transportation Research Board Meeting*, Washington D.C., USA, 2010, p. 17p.

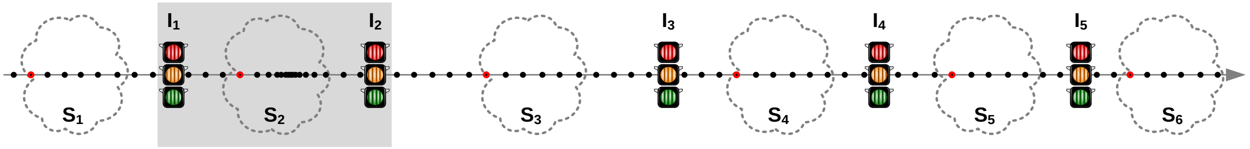
[12] Y. Malinovskiy, U.-K. Lee, Y.-J. Wu, and Y. Wang, “Investigation of Bluetooth-Based Travel Time Estimation Error on a Short Corridor,” in *Proceedings of the 90th Annual Transportation Research Board Meeting*, Washington D.C., USA, 2011, p. 19p.

[13] “PostgreSQL: About.” [Online]. Available: http://www.postgresql.org/about/. [Accessed: 18-Oct-2011].

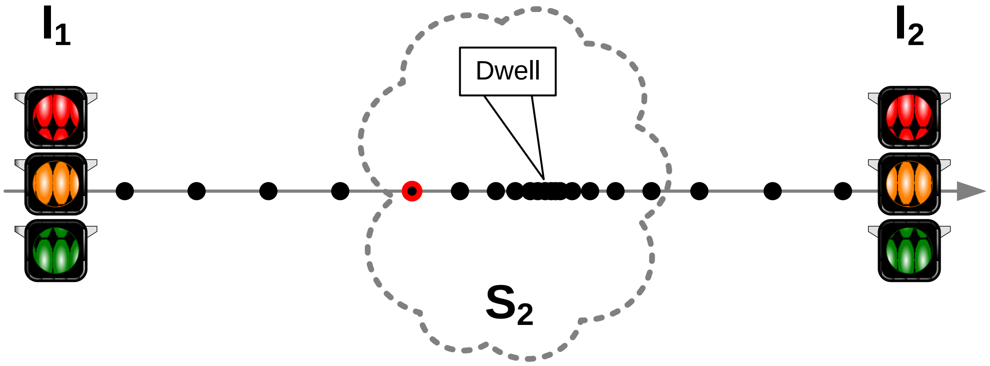
[14] “PostgreSQL: Documentation: Manuals: PostgreSQL 9.1: Views.” [Online]. Available: http://www.postgresql.org/docs/current/interactive/tutorial-views.html. [Accessed: 19-Oct-2011].

LIST OF FIGURES

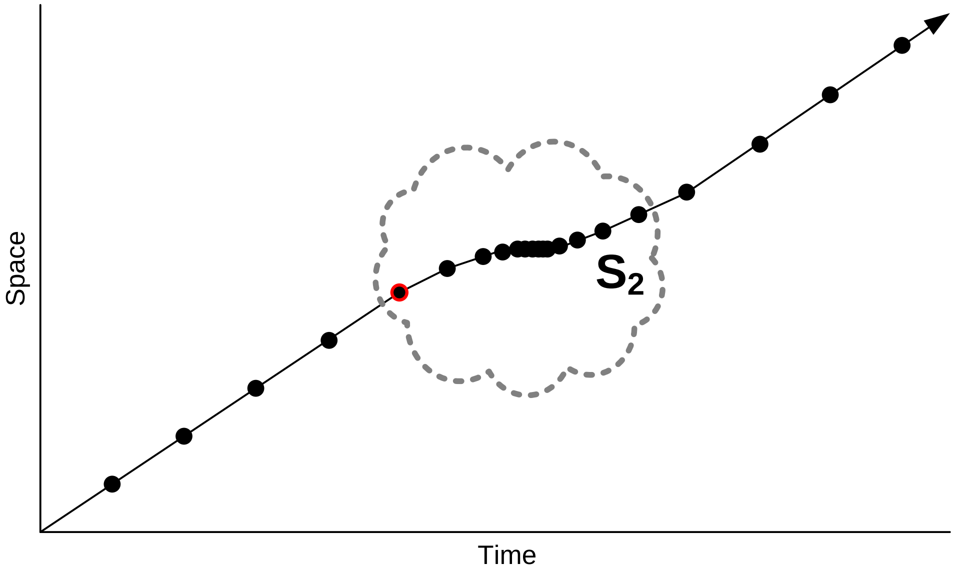
LIST OF TABLES



(a) Corridor Probe Trajectory with Earliest Observation Highlighted



(b) Segment Probe Trajectory with Earliest Observation Highlighted



(c) Time-Space Diagram for Vehicle Trajectory through Variable Speed Segment with Stop

Figure 1: Spatial and Temporal Considerations for Bluetooth Data Acquisition

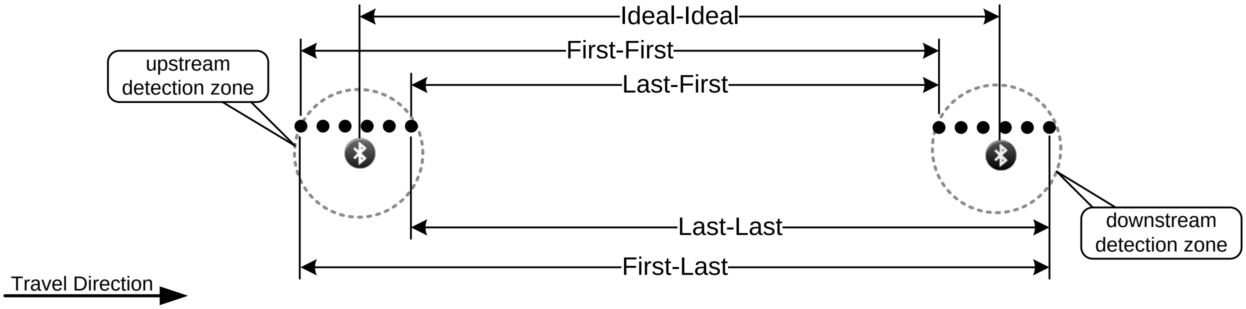
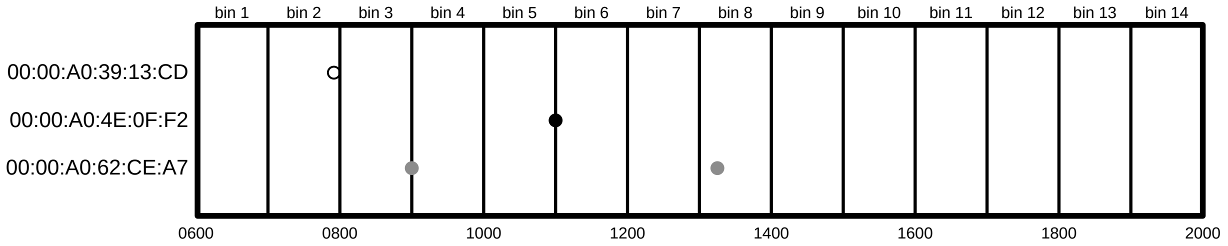
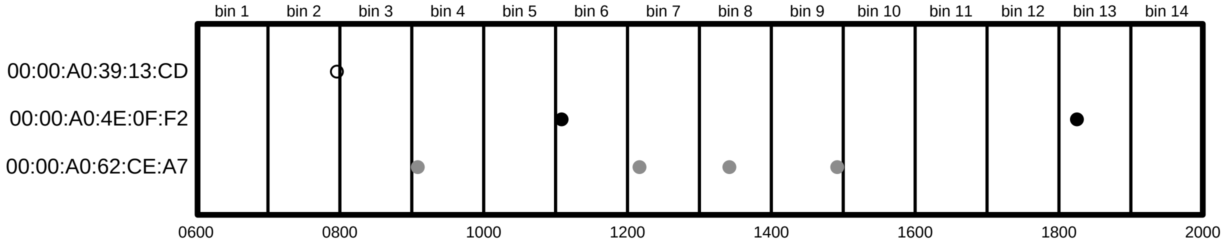


Figure 2: Pattern Matching Options for Probe Data

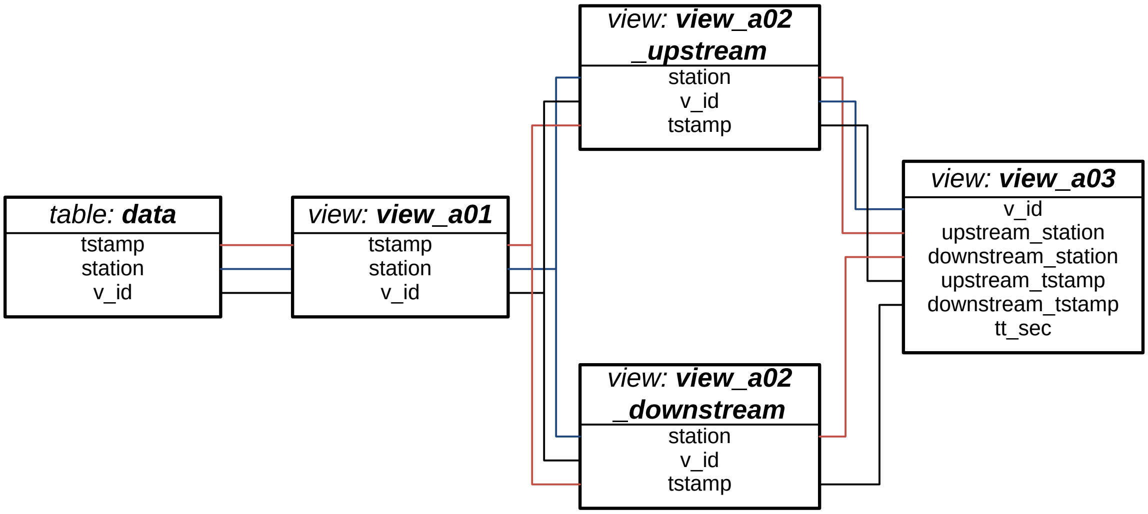


(a) Vehicle ID Observations at Station 1



(b) Vehicle ID Observations at Station 2

Figure 3: Bluetooth Observation Profiles



(a) Entity Relationship Diagram

```
SELECT
data.station,
data.v_id,
data.tstamp
FROM data
WHERE
(data.station = 's1' OR
data.station = 's2') AND
data.tstamp >= '2011-07-13
06:00:00' AND
data.tstamp < '2011-07-13
20:00:00';
```

(b) view_a01 SQL statement

```
SELECT
view_a01.station,
view_a01.v_id,
view_a01.tstamp
FROM view_a01
WHERE view_a01.station = 's1';
```

(c) view_a02_upstream SQL statement

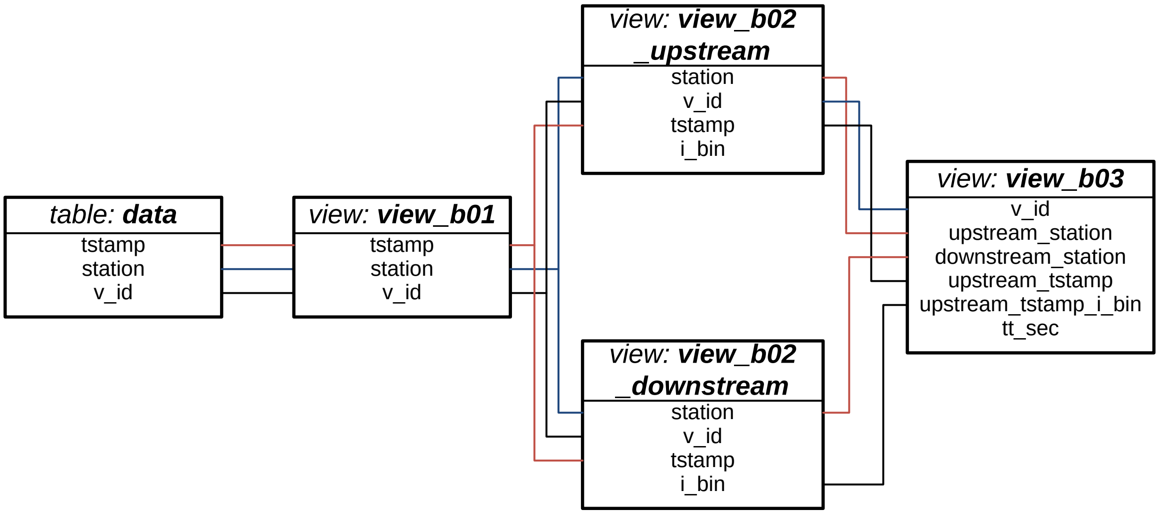
```
SELECT
view_a01.station,
view_a01.v_id,
view_a01.tstamp
FROM view_a01
WHERE view_a01.station = 's2';
```

(d) view_a02_downstream SQL statement

```
SELECT
view_a02_upstream.v_id,
view_a02_upstream.station AS upstream_station,
view_a02_downstream.station AS downstream_station,
view_a02_upstream.tstamp AS upstream_tstamp,
view_a02_downstream.tstamp AS downstream_tstamp,
(date_part('epoch',view_a02_downstream.tstamp) - date_part('epoch',view_a02_upstream.tstamp)) AS tt_sec
FROM view_a02_upstream, view_a02_downstream
WHERE
view_a02_upstream.v_id = view_a02_downstream.v_id AND
view_a02_upstream.tstamp < view_a02_downstream.tstamp
ORDER BY
view_a02_upstream.v_id, view_a02_upstream.tstamp;
```

(e) view_a03 SQL statement

Figure 4: Entity Relationships and SQL Statements for Data Reduction Strategy A



(a) Entity Relationship Diagram

```
SELECT
data.station,
data.v_id,
data.tstamp
FROM data
WHERE
(data.station = 's1' OR
data.station = 's2') AND
data.tstamp >= '2011-07-13
06:00:00' AND
data.tstamp < '2011-07-13
20:00:00';
```

(b) view_b01 SQL statement

```
SELECT
view_b01.station,
view_b01.v_id,
min(view_b01.tstamp) AS tstamp,
trunc(date_part('epoch',
view_b01.tstamp) / 3600) AS
i_bin
FROM view_b01
WHERE view_b01.station = 's1'
GROUP BY
view_b01.station,
view_b01.v_id,
i_bin;
```

(c) view_b02_upstream SQL statement

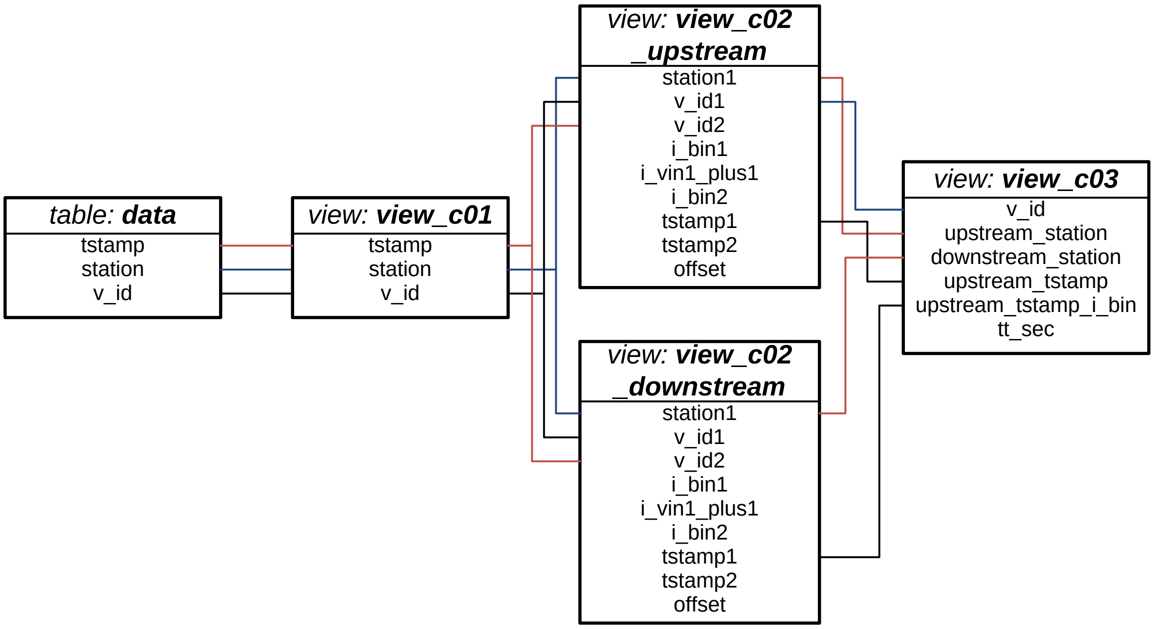
```
SELECT
view_b01.station,
view_b01.v_id,
min(view_b01.tstamp) AS tstamp,
trunc(date_part('epoch',
view_b01.tstamp) / 3600) AS i_bin
FROM view_b01
WHERE view_b01.station = 's2'
GROUP BY
view_b01.station,
view_b01.v_id,
i_bin;
```

(d) view_b02_downstream SQL statement

```
SELECT
view_b02_upstream.v_id,
view_b02_upstream.station AS upstream_station,
view_b02_downstream.station AS downstream_station,
view_b02_upstream.tstamp AS upstream_tstamp,
view_b02_upstream.i_bin AS upstream_tstamp_i_bin,
min(date_part('epoch', view_b02_downstream.tstamp) - date_part('epoch', view_b02_upstream.tstamp)) AS
tt_sec
FROM view_b02_upstream, view_b02_downstream
WHERE view_b02_upstream.v_id = view_b02_downstream.v_id AND view_b02_upstream.tstamp <
view_b02_downstream.tstamp
GROUP BY
view_b02_upstream.v_id,
view_b02_upstream.station,
view_b02_downstream.station,
view_b02_upstream.tstamp,
view_b02_upstream.i_bin;
```

(e) view_b03 SQL statement

Figure 5: Entity Relationships and SQL Statements for Data Reduction Strategy B



(a) Entity Relationship Diagram

```
SELECT
data.station,
data.v_id,
data.tstamp
FROM data
WHERE
(data.station = 's1' OR
data.station = 's2') AND
data.tstamp >= '2011-07-13 06:00:00' AND
data.tstamp < '2011-07-13 20:00:00';
```

(b) view_c01 SQL statement

Figure 6: Entity Relationships and SQL Statements for Data Reduction Strategy C – Part 1

```
WITH bin1 AS (
SELECT view_c01.station,
view_c01.v_id,
min(view_c01.tstamp) AS tstamp,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
AS i_bin,
trunc(date_part('epoch', view_c01.tstamp) / 3600) +
1 AS i_bin_plus_1
FROM view_c01
WHERE view_c01.station = 's1'
GROUP BY view_c01.station,
view_c01.v_id,
trunc(date_part('epoch', view_c01.tstamp) / 3600),
trunc(date_part('epoch', view_c01.tstamp) / 3600) +
1
),
bin2 AS (
SELECT view_c01.station,
view_c01.v_id,
min(view_c01.tstamp) AS tstamp,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
AS i_bin
FROM view_c01
WHERE view_c01.station = 's1'
GROUP BY view_c01.station,
view_c01.v_id,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
)
SELECT
bin1.station AS station1,
bin1.v_id AS v_id1,
bin2.v_id AS v_id2,
bin1.i_bin AS i_bin1,
bin1.i_bin_plus_1 AS i_bin1_plus_1,
bin2.i_bin AS i_bin2,
bin1.tstamp AS tstamp1,
bin2.tstamp AS tstamp2,
(date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp)) AS offset
FROM
bin1 LEFT JOIN bin2 ON
bin1.v_id = bin2.v_id AND
bin1.i_bin_plus_1 = bin2.i_bin
WHERE
date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp) > 3600 OR
date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp) IS NULL;
```

(a) view_c02_upstream SQL statement

```
WITH bin1 AS (
SELECT view_c01.station,
view_c01.v_id,
min(view_c01.tstamp) AS tstamp,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
AS i_bin,
trunc(date_part('epoch', view_c01.tstamp) / 3600) +
1 AS i_bin_plus_1
FROM view_c01
WHERE view_c01.station = 's2'
GROUP BY view_c01.station,
view_c01.v_id,
trunc(date_part('epoch', view_c01.tstamp) / 3600),
trunc(date_part('epoch', view_c01.tstamp) / 3600) +
1
),
bin2 AS (
SELECT view_c01.station,
view_c01.v_id,
min(view_c01.tstamp) AS tstamp,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
AS i_bin
FROM view_c01
WHERE view_c01.station = 's2'
GROUP BY view_c01.station,
view_c01.v_id,
trunc(date_part('epoch', view_c01.tstamp) / 3600)
)
SELECT
bin1.station AS station1,
bin1.v_id AS v_id1,
bin2.v_id AS v_id2,
bin1.i_bin AS i_bin1,
bin1.i_bin_plus_1 AS i_bin1_plus_1,
bin2.i_bin AS i_bin2,
bin1.tstamp AS tstamp1,
bin2.tstamp AS tstamp2,
(date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp)) AS offset
FROM
bin1 LEFT JOIN bin2 ON
bin1.v_id = bin2.v_id AND
bin1.i_bin_plus_1 = bin2.i_bin
WHERE
date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp) > 3600 OR
date_part('epoch', bin2.tstamp) -
date_part('epoch', bin1.tstamp) IS NULL;
```

(b) view_c02_downstream SQL statement

```
SELECT
view_c02_upstream.v_id1 AS v_id,
view_c02_upstream.station1 AS upstream_station,
view_c02_downstream.station1 AS downstream_station,
view_c02_upstream.tstamp1 AS upstream_tstamp,
view_c02_upstream.i_bin1 AS upstream_tstamp_i_bin,
min(date_part('epoch',view_c02_downstream.tstamp1) - date_part('epoch',view_c02_upstream.tstamp1)) AS
tt_sec
FROM view_c02_upstream,view_c02_downstream
WHERE view_c02_upstream.v_id1 = view_c02_downstream.v_id1 AND view_c02_upstream.tstamp1 <
view_c02_downstream.tstamp1
GROUP BY
view_c02_upstream.v_id1,
view_c02_upstream.station1,
view_c02_downstream.station1,
view_c02_upstream.tstamp1,
view_c02_upstream.i_bin1;
```

(c) view_c03 SQL statement

Figure 7: SQL Statements for Data Reduction Strategy C - Part 2

Line	v_id	station	tstamp
1	00:00:A0:11:59:18	s2	7/13/11 14:06:57
2	00:00:A0:11:59:18	s1	7/13/11 14:12:23
3	00:00:A0:20:4A:D6	s2	7/13/11 10:57:15
4	00:00:A0:28:6E:B2	s2	7/13/11 8:31:31
5	00:00:A0:28:6E:B2	s2	7/13/11 8:31:26
6	00:00:A0:28:94:58	s2	7/13/11 16:49:35
7	00:00:A0:28:94:58	s2	7/13/11 16:49:30
8	00:00:A0:28:96:25	s2	7/13/11 6:39:20
9	00:00:A0:28:96:25	s1	7/13/11 6:44:42
10	00:00:A0:39:13:CD	s2	7/13/11 7:55:22
11	00:00:A0:39:13:CD	s2	7/13/11 7:55:26
12	00:00:A0:39:13:CD	s1	7/13/11 7:50:09
13	00:00:A0:39:13:CD	s1	7/13/11 7:50:12
14	00:00:A0:39:D0:EC	s1	7/13/11 11:43:30
15	00:00:A0:3E:05:00	s1	7/13/11 13:35:50
16	00:00:A0:41:7A:A0	s1	7/13/11 9:28:57
17	00:00:A0:41:7A:A0	s1	7/13/11 15:15:57
18	00:00:A0:41:7A:A0	s1	7/13/11 15:36:21
19	00:00:A0:48:7F:96	s2	7/13/11 7:30:11
20	00:00:A0:48:7F:96	s2	7/13/11 7:30:09
21	00:00:A0:48:7F:96	s2	7/13/11 17:47:11
22	00:00:A0:48:7F:96	s2	7/13/11 17:47:06
23	00:00:A0:48:7F:96	s1	7/13/11 7:38:34
24	00:00:A0:4D:47:9C	s1	7/13/11 7:38:30

Table 1: Subset of Results from view_a01 for Data Reduction Strategy A

Line	v_id	upstream_station	downstream_station	upstream_tstamp	downstream_tstamp	tt_sec
1	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:09	7/13/11 7:55:22	313
2	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:09	7/13/11 7:55:26	317
3	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:12	7/13/11 7:55:22	310
4	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:12	7/13/11 7:55:26	314
5	00:00:A0:48:7F:96	s1	s2	7/13/11 7:38:34	7/13/11 17:47:06	36512
6	00:00:A0:48:7F:96	s1	s2	7/13/11 7:38:34	7/13/11 17:47:11	36517
7	00:00:A0:4E:0F:F2	s1	s2	7/13/11 10:59:59	7/13/11 18:19:34	26375
8	00:00:A0:4E:0F:F2	s1	s2	7/13/11 10:59:59	7/13/11 11:04:36	277
9	00:00:A0:4E:0F:F2	s1	s2	7/13/11 10:59:59	7/13/11 11:04:39	280
10	00:00:A0:4E:0F:F2	s1	s2	7/13/11 11:00:03	7/13/11 11:04:36	273
11	00:00:A0:4E:0F:F2	s1	s2	7/13/11 11:00:03	7/13/11 11:04:39	276
12	00:00:A0:4E:0F:F2	s1	s2	7/13/11 11:00:03	7/13/11 18:19:34	26371
13	00:00:A0:4F:02:E5	s1	s2	7/13/11 19:14:37	7/13/11 19:20:01	324
14	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	7/13/11 9:05:29	303
15	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	7/13/11 15:55:55	24929
16	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	7/13/11 12:11:07	11441
17	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	7/13/11 13:29:23	16137
18	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	7/13/11 15:55:50	24924
19	00:00:A0:62:CE:A7	s1	s2	7/13/11 13:24:14	7/13/11 15:55:55	9101
20	00:00:A0:62:CE:A7	s1	s2	7/13/11 13:24:14	7/13/11 13:29:23	309
21	00:00:A0:62:CE:A7	s1	s2	7/13/11 13:24:14	7/13/11 15:55:50	9096
22	00:00:A0:65:BA:09	s1	s2	7/13/11 17:41:30	7/13/11 17:48:02	392
23	00:00:A0:65:BA:09	s1	s2	7/13/11 17:41:30	7/13/11 17:48:11	401
24	00:00:A0:65:BA:09	s1	s2	7/13/11 17:41:30	7/13/11 17:48:07	397

Table 2: Subset of Results from view_a03 for Data Reduction Strategy A

Line	v_id	upstream_station	downstream_station	upstream_tstamp	upstream_tstamp_i_bin	tt_sec
1	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:09	364048	313
2	00:00:A0:48:7F:96	s1	s2	7/13/11 7:38:34	364048	36512
3	00:00:A0:4E:0F:F2	s1	s2	7/13/11 10:59:59	364051	277
4	00:00:A0:4E:0F:F2	s1	s2	7/13/11 11:00:03	364052	273
5	00:00:A0:4F:02:E5	s1	s2	7/13/11 19:14:37	364060	324
6	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	364050	303
7	00:00:A0:62:CE:A7	s1	s2	7/13/11 13:24:14	364054	309
8	00:00:A0:65:BA:09	s1	s2	7/13/11 17:41:30	364058	392
9	00:00:A0:66:5D:8A	s1	s2	7/13/11 10:06:31	364051	322
10	00:00:A0:6A:32:4B	s1	s2	7/13/11 13:15:23	364054	341
11	00:00:A0:6F:F8:3B	s1	s2	7/13/11 11:45:41	364052	319
12	00:00:A0:7B:8A:7F	s1	s2	7/13/11 18:27:59	364059	3886
13	00:00:A0:7B:C1:1C	s1	s2	7/13/11 6:16:27	364047	42882
14	00:00:A0:7B:C1:1C	s1	s2	7/13/11 18:06:10	364059	299
15	00:02:72:AB:19:25	s1	s2	7/13/11 18:22:32	364059	298
16	00:03:7A:C8:AF:43	s1	s2	7/13/11 7:57:52	364048	338
17	00:03:7A:EF:95:33	s1	s2	7/13/11 8:30:49	364049	4722
18	00:05:4F:03:AA:46	s1	s2	7/13/11 14:22:44	364055	305
19	00:05:4F:04:2F:C8	s1	s2	7/13/11 14:13:55	364055	278
20	00:05:4F:04:69:49	s1	s2	7/13/11 7:37:16	364048	325
21	00:05:4F:04:6C:62	s1	s2	7/13/11 19:36:14	364060	308
22	00:05:4F:04:89:BD	s1	s2	7/13/11 18:44:24	364059	320
23	00:05:4F:04:FD:D8	s1	s2	7/13/11 14:40:13	364055	278
24	00:05:4F:07:EB:FB	s1	s2	7/13/11 6:41:24	364047	25942

Table 3: Subset of Results from view_b03 for Data Reduction Strategy B

Line	v_id	upstream_station	downstream_station	upstream_tstamp	upstream_tstamp_i_bin	tt_sec
1	00:00:A0:39:13:CD	s1	s2	7/13/11 7:50:09	364048	313
2	00:00:A0:48:7F:96	s1	s2	7/13/11 7:38:34	364048	36512
3	00:00:A0:4E:0F:F2	s1	s2	7/13/11 11:00:03	364052	273
4	00:00:A0:4F:02:E5	s1	s2	7/13/11 19:14:37	364060	324
5	00:00:A0:62:CE:A7	s1	s2	7/13/11 9:00:26	364050	303
6	00:00:A0:62:CE:A7	s1	s2	7/13/11 13:24:14	364054	309
7	00:00:A0:65:BA:09	s1	s2	7/13/11 17:41:30	364058	392
8	00:00:A0:66:5D:8A	s1	s2	7/13/11 10:06:31	364051	322
9	00:00:A0:6A:32:4B	s1	s2	7/13/11 13:15:23	364054	341
10	00:00:A0:6F:F8:3B	s1	s2	7/13/11 11:45:41	364052	319
11	00:00:A0:7B:8A:7F	s1	s2	7/13/11 18:27:59	364059	3886
12	00:00:A0:7B:C1:1C	s1	s2	7/13/11 6:16:27	364047	42882
13	00:00:A0:7B:C1:1C	s1	s2	7/13/11 18:06:10	364059	299
14	00:02:72:AB:19:25	s1	s2	7/13/11 18:22:32	364059	298
15	00:03:7A:C8:AF:43	s1	s2	7/13/11 7:57:52	364048	338
16	00:03:7A:EF:95:33	s1	s2	7/13/11 8:30:49	364049	6036
17	00:05:4F:03:AA:46	s1	s2	7/13/11 14:22:44	364055	305
18	00:05:4F:04:2F:C8	s1	s2	7/13/11 14:13:55	364055	278
19	00:05:4F:04:69:49	s1	s2	7/13/11 7:37:16	364048	325
20	00:05:4F:04:6C:62	s1	s2	7/13/11 19:36:14	364060	308
21	00:05:4F:04:89:BD	s1	s2	7/13/11 18:44:24	364059	320
22	00:05:4F:04:FD:D8	s1	s2	7/13/11 14:40:13	364055	278
23	00:05:4F:07:EB:FB	s1	s2	7/13/11 6:41:24	364047	25942
24	00:05:4F:07:EB:FB	s1	s2	7/13/11 13:48:29	364054	317

Table 4: Subset of Results from view_c03 for Data Reduction Strategy C