2006-22

Final Report

Development of Efficient Integrated Data Archival/Retrieval Model for R/WIS, RTMS, and Loop Traffic Data

Research

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
MN/RC-2006-22		

4. Title and Subtitle Development of Efficient Integrated Data Archival/Retrieval Model for R/WIS, RTMS, and Loop Traffic Data	5. Report Date June 2006
	6.
7. Author(s) Taek Mu Kwon, Ph.D	8. Performing Organization Report No.
9. Performing Organization Name and Address University of Minnesota Duluth Department of Electrical and Computer Engineering 1023 Unversity Drive Duluth, Minnesota 55812	10. Project/Task/Work Unit No.
	11. Contract (C) or Grant (G) No. (C) 81655 (wo) 142
12. Sponsoring Organization Name and Address Minnesota Department of Transportation Research Services Section 395 John Ireland Boulevard, Mail Stop 330 St. Paul, Minnesota 55155	13. Type of Report and Period Covered Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes
http://www.lrrb.org/PDF/200622.pdf

16. Abstract (Limit: 200 words)

This report describes a new data warehouse model developed for integrating Road Weather Information System (R/WIS) and traffic data and the prototype implemented. The building blocks of the prototype include data aggregation methods from sensors, a data archiving system, and multi-user data access and retrieval environments through a network. This new data warehouse model seamlessly integrates the heterogeneous nature of R/WIS and traffic data. The key to this data model was utilization of a network storage model referred to as a parallel First-In-First-Out (FIFO) data storage where various sensor data are deposited as they are aggregated while different types of data-consuming modules obtain data without an explicit protocol requirement. For the prototype implementation, four different data aggregation methods from traffic and R/WIS sources were used to demonstrate that diverse data types and collection methods could be seamlessly integrated together. As an application of this data warehouse, weather impact on traffic flow was studied by retrieving traffic data under various atmospheric and pavement conditions, and the results are included. It was noticed that R/WIS provides a significant advantage over the traditional National Weather Service data in learning detailed location specific weather and pavement conditions from which weather impact on traffic flow could be accurately analyzed.

17. Document Analysis/Descriptors	18. Availability Statement
Data aggregation and warehousing, R/WIS and traffic data, Weather impact on traffic	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	59	

Development of Efficient Integrated Data Archival/Retrieval Model for R/WIS, RTMS, and Loop Traffic Data

Final Report

Prepared by:
Taek M. Kwon

Transportation Data Research Laboratory (TDRL)
Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

Published by:

Minnesota Department of Transportation
Research Services Section, MS 330
395 John Ireland Boulevard
St. Paul, MN 55155

June 2006

Acknowledgements

The author would like to acknowledge several people at Mn/DOT and UMD students who contributed to this project. Mr. Edward Fleege at Mn/DOT District-1 originally helped to formulate this project and served as a technical liaison. After his retirement from Mn/DOT, Ms. Roberta Dwyer kindly took over his role and provided much assistance to this project. District-1 electricians installed RTMS sensors and cabinets for this project, so their efforts are recognized. Several undergraduate and graduate students at UMD worked on this project. In particular, Dan Rogahn worked on interface programming on RTMS and RTC commands and developed XML structures for traffic and R/WIS data. Jason Michelle wrote several modules for data acquisition from R/WIS and traffic data archives, and created a relational database. Lalit Nookala improved the relational database, created an active web interface, and generated the weather impact data shown in this report.

Table of Contents

List of Figures

List of Tables

Executive Summary

In the past, Road Weather Information System (R/WIS) and traffic data have mostly been managed in isolation, thus underutilizing the benefits attainable by integrating both types of data. Researchers believe that historical and real-time traffic data integrated with R/WIS data can lead to a better understanding of traffic by correlating road weather conditions to the traffic data. The objective of this project was to develop a new data warehouse model from the ground up starting from sensor data and building up to applications to demonstrate the benefits of data integration. Based on this objective, a new data warehouse model was developed to seamlessly integrate the heterogeneous nature of R/WIS and traffic data. This report describes the data model developed and the prototype implemented. The functions of the prototype include a data aggregation model from sensors, a data archiving method, and multi-user data access and retrieval environments through a network. The key to this data model was utilization of a network storage model referred to as a parallel First-In-First-Out (FIFO) data storage where various sensor data are deposited as they are aggregated while different types of data-consuming modules obtain data without an explicit protocol requirement. For the prototype implementation, four different data aggregation methods from traffic and R/WIS sources were used to demonstrate that diverse data types and collection methods could be seamlessly integrated together. As an application of this data warehouse, the effect of pavement conditions on traffic flow was studied, and the results are included.

CHAPTER 1

INTRODUCTION

In most state transportation departments, Road Weather Information System (R/WIS) data is commonly managed by a maintenance office and traffic data by a traffic management center (TMC). Initial investment, maintenance, and usages for different purposes of the two types of data commonly instill such separate data management practices. On the other hand, most transportation engineers and decision makers agree on the benefits of integrating the R/WIS and traffic data into one resource. For example, inclement weather conditions often significantly affect traffic flow and capacity. If the effect on traffic flow by weather conditions (using traffic and R/WIS data) is measurable, such measurements could be incorporated into traffic control strategies and travel advisory systems, which would lead to more accurate traffic control and better travel information. From the perspective of winter road maintenance, ice/snow removal schedules and route decisions could be more precisely managed or optimized by incorporating real-time and historic traffic flow data. These are just two apparent examples, but perhaps many other direct and indirect benefits could be drawn from the integration of traffic and R/WIS data. Presently, the data sharing is only available at a web dissemination level in Mn/DOT. To attain the full benefits of data integration, it is desirable that the two types of data are integrated as a single unified data system that allows fast retrieval of both types of data. This need was the motivation for this project.

Although the concept of integrating R/WIS and traffic data is straightforward, the details in the system design present many challenges. For example, even within the R/WIS data, Remote Processing Units (RPUs) from different manufacturers have different data types, formats, accuracies, and protocols [2]. Time interval differences and synchronization are another difficult problem. Presently, Mn/DOT R/WIS data is recorded every 10 minutes while TMC loop-detector data is recorded every 30 seconds. This interval difference makes one-to-one mapping or correlation difficult. Another difficult challenge is in dealing with the time synchronization problems within the data already collected, i.e., the timestamps of the data collected from different locations are often neither synchronized nor accurate. For example, timestamp 10:00 AM in one RPU could be indeed 11:00 AM in another RPU, and there is no easy way of verifying which time is correct. Reliability of data collection devices or sensors themselves is another problem that affects data quality. Data is often lost for an extended period due to reliability problems in hardware and software. Moreover, even if the sensor hardware and data collection software work without failures, the large amount of the continuous stream of data presents another challenge [1,4]. A continuous stream of data from sensors means that any stop in data handling instruments could lead to loss of data, and it is difficult to make any instrument work indefinitely. Moreover, efficient data archiving and database techniques are needed to deal with the problems with large-scaled data [4].

According to the data experience at the Transportation Data Research Laboratory (TDRL), some of the problems mentioned above cannot be solved by improving only a few modules or components. However, well-designed building blocks of a data warehouse and intelligent interaction among them could overcome many of the problems mentioned and could create a reliable data system. This project was intended to provide a solution in that perspective for the end goal of building a reliable and efficient data integration system. Moreover, to demonstrate practicality from the theoretical model, a prototype data warehouse that integrates both types of data was constructed in this project.

The project used several different data sources, different data types, and data communication methods to explore the diversity effects in the data. More specifically, four software modules ran simultaneously to handle four different types of data sources, from which data was deposited to a shared pool of network storage called the FIFO (First-in-First-Out) storage. The FIFO data was then consumed by various software modules to create data warehouse functions that included a relational database and an archive management system. Based on the constructed data warehouse, several applications were developed for demonstrating utilization of the integration. This report provides the basic concept for the data architecture developed and the actual prototype implementation of the data warehouse model.

CHAPTER 2

INTEGRATED TRANSPORTATION SENSOR DATA ARCHITECTURE

One of the important characteristics of transportation sensor data (TSD) that must be considered for the efficient design of data model or management is the "continuous flow of data until the sensor is retired from the system." Once any sensor is powered up, it is supposed to continuously produce data, and any stoppage of its operation results in a loss of data. Another characteristic of TSD is that the number of sensors tends to continuously change (increase) over time. For example, statewide transportation sensor networks typically start with a small number of sensors but the number typically increases to a large number. Today, operating tens of thousands of sensors to monitor traffic or road weather conditions is common. The two characteristics, continuous and many, typically accumulate to a very large amount of data, and the data would eventually fill up any finite storage. Therefore, careful management of storage is essential since any mismanagement can lead to loss of a large amount of data.

Another characteristic of TSD is the inclusion of diverse types of data and sampling rates. For example, the Mn/DOT statewide R/WIS produces 37 different types of parameters that include non-numeric values. The challenge is that all of the diversity in data types must be handled in parallel through several different means of communication links from thousands of sensors. In addition, this diversity changes with time due to introduction of new sensing technologies and retirement of old technologies.

The integrated TSD data warehouse model in this project was developed by considering the characteristics of TSD mentioned above, and Figure 1 describes its conceptual model using a block diagram. This model includes a complete path of data flow from the sensor level to applications. At the top, it starts with an environment that we wish to observe, as shown in Figure 1. The role of sensors is then to monitor a specific part of the environment, and at the very basic level it is simply a converter that converts a property of the environment to an electrical signal. However, the electric signal does not become a datum until a data acquisition system converts it. This data flow relation is shown in the model (Figure 1) by a connection from the environment to sensors and then another connection to data acquisition systems.

Most data acquisition systems are made to hold only a limited amount of data since its role is in converting the electrical signal to a datum. As mentioned before, the data produced by a data acquisition system is a continuous stream. A common type of technique that allows management of a data stream is referred to as a buffer in which data is temporarily held to neutralize the discrepancy of the processing speed between the data source and consumer.

The proposed data model includes a FIFO storage in which the old data are removed as new data arrives. In addition to the data buffering, another role required for the FIFO storage in the data model is to start data conversion to create a form that can be uniformly understood by the next layer of processes. This is because it is easier and much efficient when the data format is unified at an early stage. Since sensor data is coming from many different locations, this storage is connected through a network and shared by data acquisition systems. It should work like an on-line shared savings account in which deposits can be made from many different locations while the activity and data can be read from any location. In addition, it should be able to accommodate a diversity of sensor data types and sampling periods. For this requirement, the data format of FIFO is specifically defined, which is described in a later chapter. The data retention period of the FIFO storage should depend on the completion of the archiving task and the data arrival rate from the data acquisition systems. In general, once the FIFO data is securely archived by the archiving module, the data could then be safely removed without the concern of any loss. At the next level, the temporary data in the FIFO storage is consumed by three data management modules, i.e., relational database management system (RDBMS), XML (Extensible Markup Language) converter, and unified TSD format (UTSDF) archiving module [4] (see Figure 1). These three modules satisfy three different types of data needs, i.e., real time, near-term, and long-term data needs. One of the real-time needs would be data dissemination through a web service in real time. For web applications, XML is a widely accepted format and could be used for data communication. On the other hand, XML is composed of uncompressed ASCII codes and adds a large amount of overhead to the original data, which would require a large bandwidth in communication. Therefore, XML should only be used for relatively small size data applications in TSD. RDBMS is another widely used form of storage and can effectively handle complex types of data query. However, RDBMS also adds significant amount of overhead to the original data, and thus it is not a cost effective method for storing a very large quantity of data such as a statewide TSD [3]. In the proposed model, RDBMS is only used for saving data up to about one year, and it supports applications requiring mid-range data size.

The third component is the archiving module. Archiving is required for the long-term storage needs in TSD, typically tens or hundreds of years. Since such a long period will eventually lead to a huge size of storage requirement, it is imperative to have an efficient, compact data format and very reliable data storage. The compact format should result in a small file size, and should allow easy exchanges of data. Regarding the choice of data format, it should be easily decodable and should not require any special hardware or software tools. More specifically, since the data has to be used for the indefinite future, the rapid changes of technologies should not make the format obsolete. These requirements are satisfied by the UTSDF method developed by TDRL, and was used for this project. Once UTSDF archives are created, the archived files can be used for large scale multi-year data studies, or can be processed and fed into another RDBMS from which complex queries can be made from the data [4]. These relations are shown in the right side of the data flow in Figure 1.

The integrated TSD architecture described in this chapter constitutes the frame work of the prototype data system developed in this project. The main key of this architecture is in the sharable network FIFO storage from which all sorts of diversity in sensors are merged as a pool of data that could be understood by many other modules. It is a simple buffer concept but was shown effective in handling many sources of real-time sensor data.

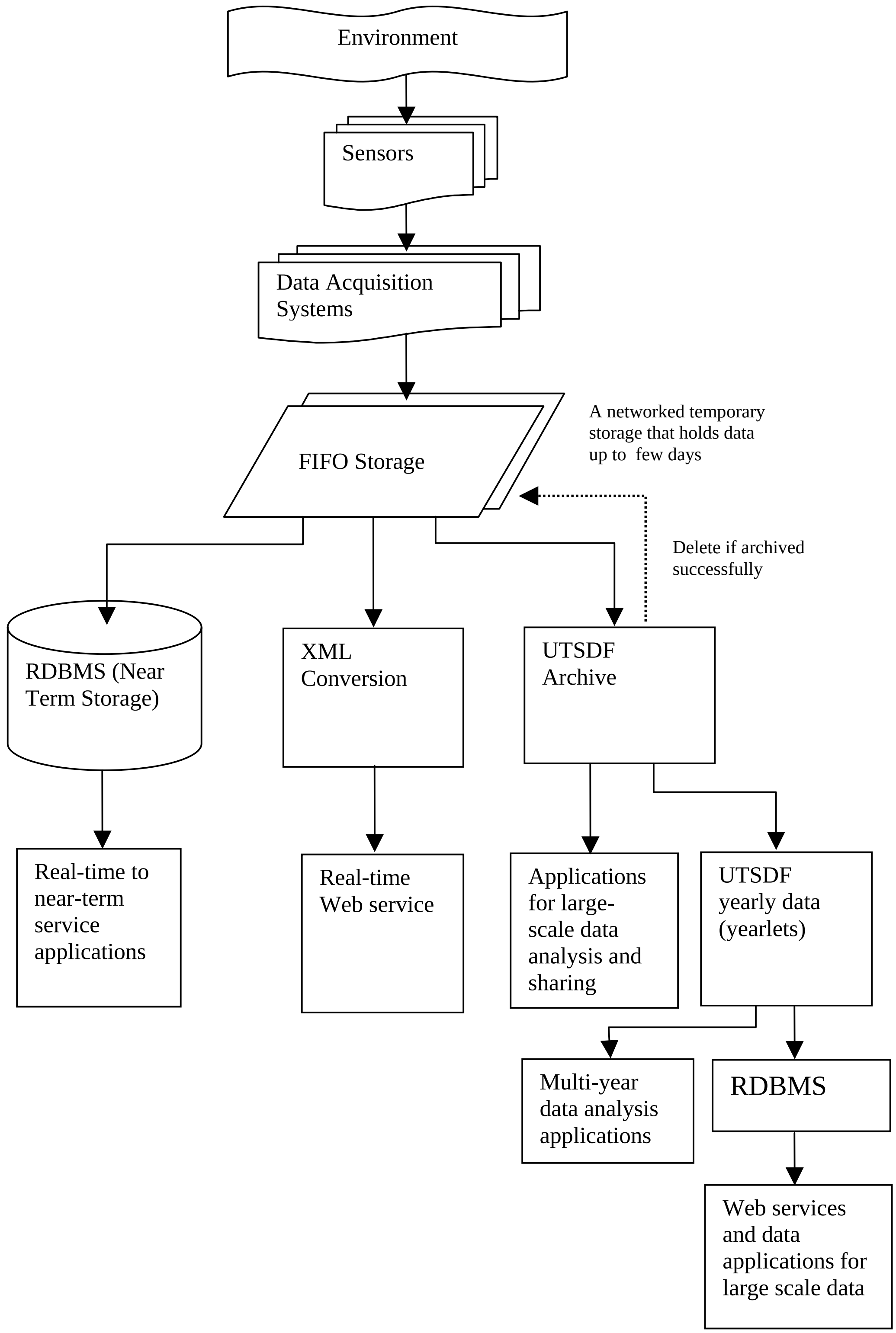


Figure 2.1: Large Scale TSD Integration Model

CHAPTER 3

IMPLEMENTATION

This chapter describes how the prototype data warehouse was implemented based on the data model described in the previous chapter. The structure and function of each module is described.

3.1 Brief History

At the start of this project, the statewide R/WIS in Mn/DOT was in a planning stage, but a small-scaled experimental R/WIS was available from District-1 and the Metro area. This project was intended to use the three existing remote processing unit (RPU) sites in District-1 at that time. These sites were later integrated to the statewide R/WIS. The three sites recommended by Mn/DOT District-1 were (1) Garfield Ave. in Duluth, (2) Thompson Hill in Nopeming, and (3) Atkinson Bridge at I-35. Each of these sites had an R/WIS RPU but did not have traffic sensors. To include traffic data needs in those sites, Radar Traffic Monitoring System (RTMS) units, which are microwave-based non-intrusive sensors, were installed. Unfortunately, the RPU at the Atkinson bridge had only a few R/WIS sensors and did not fully work most of the time. In addition, RTMS installation was significantly delayed due to malfunction problems, limited installation resources at District-1, and other construction at the site. Meanwhile, installation of the statewide R/WIS was completed and became fully operational. Due to the recurring problems, the Atkinson site was dropped from the final selection of sites in this study. At the end, a total of eight sites based on the statewide R/WIS, which include two sites from District-1 and six sites from the Metro district, were selected.

3.2 Data Sources

The selection of the sites in this project was made based on the availability of traffic sensors near the R/WIS RPU in order to use both types of the data. All sites included in the Mn/DOT's statewide R/WIS were reviewed, and eight sites were found to have traffic sensors near the R/WIS site. The rest of the sites did not have traffic sensors and were not included in this study. The final selection of the sites is summarized in Table 1. For the Twin Cities' area, traffic data is available from the Regional Traffic Management Center (RTMC), a part of Mn/DOT. Presently, RTMC traffic data is exported to the TDRL (Transportation Data Research Laboratory) where the data is archived and the prototype runs. A complete set of Mn/DOT statewide R/WIS data is also presently archived at TDRL.

As shown in Table 1, a total of six sites from the Twin Cities area (Metro District) were selected. The limit on the number of stations was due to unavailability of R/WIS sites at the locations where RTMC is collecting traffic data. The selected sites in the Twin Cities include Mississippi River, Minnetonka Blvd, Burnsville, Maple Grove, Little Canada, and the Cayuga St. Bridge. Figure 2 shows the location of the six sites selected in the Twin Cities on a map. The red-circled number corresponds with the index of Table 1. These six sites have inductive loop detectors in a close proximity of the RPU tower, and the data is available through RTMC.

Table 3.1: R/WIS Sites Selected for the Study

	Site Name	System ID	Site ID	Location Description	District
1	Mississippi River	142	0	I-35W over Mississippi River	Metro
2	Burnsville	330	85	I-35W near Exit 4B, Minnesota River	Metro
3	Maple Grove	330	86	I-94 near 494/694 Split	Metro
4	Little Canada	330	87	I-694 and I-35E	Metro
5	Cayuga St. Bridge	330	88	I-35E Mile Point 108	Metro
6	Minnetonka Blvd	330	84	I-494 & Minnetonka Blvd	Metro
7	Thompson Hill	158	0	Thompson Hill, Nopeming	District-1
8	Garfield Ave	158	1	Garfield Ave, Duluth	District-1

Figure 3.1: R/WIS and Traffic Sensor Stations in Twin Cities' Area. The red circles with numbers indicate the location of combined R/WIS and traffic sites, and the numbers correspond with the index in Table 1 and 2.

Figure 3.2: R/WIS and Traffic Sensor Stations in the Duluth Area. The red circles with numbers indicate the location of combined R/WIS and traffic sites, and the numbers correspond with the index in Table 1 and 2.

The two sites selected from District-1 are shown in Figure 3. These are the old R/WIS sites in District-1 that were used as an experimental system, and RTMC traffic detectors were added to the site for this project. Again, the numeric circles correspond with the indices of Table 1. The system ID and site ID used for each R/WIS site are the same number used by the Mn/DOT statewide R/WIS. The system ID distinguishes the type of system according to the manufacturers of RPU while the site ID distinguishes the physical location of the site.

Mn/DOT presently does not use any system ID or site ID for traffic detectors. In order to create uniformity in identifying sensors, new system IDs and site IDs were allocated for the traffic detectors used for this project. These new allocated numbers are summarized in Table 2 where the index of each site matches with Table 1. For System IDs, numeric number 1 was assigned for loop detectors, and numeric number 2 was allocated for RTMS. This allows adaptation of new sensor technology without change of the base data structure. For Site IDs, numeric number 1 was assigned for detectors at District-1, and numeric number 5 was assigned for the Metro district. These numbers correspond to the District numbers used by Mn/DOT.

Table 3.2: Traffic Detector Locations and IDs at the Selected R/WIS Sites

	Site Name	System ID	Site ID	Detector ID	Data Collection
1	Mississippi River	1	5	2224,2225,2226, 2149,2150,2151	Intranet File
2	Burnsville	1	5	1006,1007,1008 494,495,496	Intranet File
3	Maple Grove	1	5	907,908,909, 910,911,912	Intranet File
4	Little Canada	1	5	2419,2420,2421, 2426,2427,2428	Intranet File
5	Cayuga St. Bridge	1	5	2465,2466,2467, 2386,2387,2388	Intranet File
6	Minnetonka Blvd	1	5	1877,1878, 1851,1852	Intranet File
7	Thompson Hill	2	1	9000,9001,9002, 9003,9004,9005	Dialup FTP
8	Garfield Ave	2	1	9006,9007,9008,9009,9010,9011	Dialup RTC

RTMC uses a unique numeric number for each loop detector, and the same detector numbers were used for this project for all loop detectors. For each of the two sites that had RTMS, six new numbers starting with 9 were assigned. This is due to the specification of RTMS that the number of lanes, which can be detected by a single RTMS, is six. The choice of two different types of traffic sensors requires three different types of data acquisition methods. Since the data for loop detectors was already available from TDRL as a form of archived data, it was accessible through a networked (intranet) file system. For the RTMS at the Thompson Hill site, the data was aggregated through a PC at the site where an FTP service was installed through a phone line. This has an advantage that a PC can more reliably control the RTMS and also it allows us to use the built-in FTP server. From the TDRL data center, this data was downloaded through a dialup Internet connection to the site and through FTP. The Garfield Avenue site has an installation with an old style dumb modem connection due to unavailability of a PC at the site. In that site, the data is temporarily stored at an RTC (RTMS Traffic Counter) to which a computer at the TDRL data center dials up and downloads the data. The differences in data collection methods are summarized in the last column of Table 2. The details on how to install an RTMS and how to design the communication-link are described in Appendix A. This information could be used as a reference for future applications of RTMS or similar instruments.

3.3 FIFO Network Storage Design

The FIFO storage in the TSD integration model (see Figure 1) is used as a temporary storage for various data acquisition systems. It should in principle be designed as a parallel and multiple data queue in which new data generated by various sensors are fed into the queue and the oldest data are thrown out from the queue. The queued data are then retained in the FIFO storage until it is safely archived by the UTSDF module. This hold and release strategy can minimize data loss from communication failures since the data is retained until safely archived. There are two ways of implementing the FIFO storage, i.e., using RAM (Random Access Memory) and using a file system. Using RAM would allow faster access than a file system. However, a danger exists that the valuable data can be totally lost if a power outage occurs. Also, the size of RAM is limited, which could cause overflow of data if the FIFO storage grows too fast. Using a file system to implement FIFO is relatively simple, and was the choice for this project. It is implemented using timestamps specified in the file name, which are then used to form a logical FIFO storage. This approach has an advantage that the data can be organized using a folder structure when many different types of data must be stored and simplify the process of creating and maintaining a logical FIFO queue. It is also import to note that the FIFO system must be accessible through a network to provide multiple, simultaneous accesses to many data acquisition systems and data consuming modules. In this project, the FIFO storage was implemented on a network disk based on hot-swappable Redundant Array of Inexpensive Disks (RAID). RAID storages have an error correction capability, which would provide the reliability needed for the FIFO memory. Another advantage of using RAID is that its access speed is relatively fast since the data is pre-loaded to RAM (a.k.a., pre-fetch) and the disks are accessed in parallel. Another property that should be mentioned for the FIFO storage is that it allows synchronization of data without a complicated protocol as long as it keeps the data until all of the data consuming modules get the needed data. If a large memory size is allocated for the FIFO storage, data can be kept a longer period and could significantly reduce data losses caused by communication link failures.

3.3.1 Filename Structure for FIFO Storage

In order to allow multiple accesses from various data acquisition sources and data consumers, the FIFO storage consists of many small files, each of which has a timestamp. Fragmentation into many small files allows more efficient parallel access than sharing a single large file. Filenames are used to structure the FIFO storage. The timestamp in the file name identifies the order of queue memory, and it is defined as follows.

Filename format: ***SenGroup.yyyyMMdd.hhmm.txt***

The SenGroup prefix identifies the sensor group that the data belongs, and four types are used: “atmos” for atmospheric data, “sur” for surface data, “sub” for subsurface data, and “traffic” for traffic data. In addition, “yyyyMMdd.hhmm” string indicates a timestamp of the data, i.e., year, month, date, hour and minute. Examples are shown below.

atmos.20040701.13.20.txt
traffic.20040701.13.20.txt

The first line indicates that it contains the atmospheric data collected at 13:20 on July 1 in 2004.

For each group, four different folders are maintained within the FIFO folder according to the sensor group. This folder structure reduces the number of files and the examples are shown below.

FIFO\atmos
FIFO\sur
FIFO\subsur
FIFO\traffic

It should be noted that no actual FIFO function is implemented, but the FIFO is logically implemented by the timestamps indicating which data is old and which data is new.

3.3.2 Data Format for FIFO Files

Inside the FIFO files, the data must follow a specific format, which is described in this sub-section. According to different types of data, the data is classified into sensor groups in which unique sensor group IDs are assigned. Present classification is shown below.

Sensor Group ID

R/WIS Atmos = 10
R/WIS Surface = 11
R/WIS Subsurface = 12
traffic = 20

The first digit of the two digit ID indicates the main classification, i.e., 1 for R/WIS and 2 for traffic. The second digit is used for identification of subgroups. As more types of sensor groups are available in the future, further classification can be added. Organizing data into sensor group has an advantage that additional sensor types can be added or removed without disrupting the overall data structure or software implementations. This property allows unlimited changes in introduction of new types of sensors and retirement of old types.

3.3.3 Detailed Column Format of FIFO Data Table

Each FIFO file is a comma separated table in which all of the column data are entered using ASCII characters. Four different types of FIFO tables are defined based on the four different sensor groups, three for R/WIS and one for traffic data. Each sensor group has a different number of columns and field definition. The detailed column (data field) definition is listed in Tables 3-6. These column definitions follow the present data structure used by Mn/DOT for compatibility.

Table 3.3: Data Fields of R/WIS Atmospheric FIFO File

Data Field	Data Format (ASCII)	Description
Sensor Group ID	10	Sensor group such as R/WIS and Traffic
System ID	Integer	Identifies differences of system
Site ID	Integer	Identifies differences of sites
Sensor ID	Integer	If only one sensor exists, it is 0. Otherwise, uses Mn/DOT number.
Date Time	MM/dd/yyyy hh:mm:ss	Sampled data time. CST or SDST depending on the date
Air Temp	Single	Degrees C
Dew Temp	Single	Degrees C
R.H	Integer	Percentage
Wind Speed Avg	Integer	Km/hr
Wind Speed Gust	Integer	Km/hr
Wind Direction Avg	Integer	Azimuth — degrees
Wind Direction Max	Integer	Azimuth — degrees
Precip Intensity	Text	See table 9
Precip Type	Text	See table 10
Visibility	Integer	Meters
Air Pressure	Single	Millibars
Pricip Rate	Integer	Cm/hr
Precip Accum	Integer	Cm since midnight local time
10 min Solar	Single	Joules per square meter
24 hr Solar	Single	Joules per square meter
24 hr Sun	Integer	Minutes over 24 hours
Air Temp Max	Single	Degrees C
Air Temp Min	Single	Degrees C
Wet Bulb Temp	Single	Degrees C
Last Precip Start	MM/dd/yy hh:mm:ss am/pm	Time (GMT)
Last Precip End	MM/dd/yy hh:mm:ss am/pm	Time (GMT)
1 hr Precip Accum	Single	Centimeters
3 hr Precip Accum	Single	Centimeters
6 hr Precip Accum	Single	Centimeters
12 hr Precip Accum	Single	Centimeters
24 hr Precip Accum	Single	Centimeters

Table 3.4: Data fields of R/WIS surface FIFO file

Data Field	Data Format (ASCII)	Description
Sensor Group ID	11	Sensor group such as R/WIS and Traffic
System ID	Integer	Identifies differences of system
Site ID	Integer	Identifies differences of sites
Sensor ID	Integer	If only one sensor exists, it is 0. Otherwise, uses Mn/DOT number.
Date Time	MM/dd/yyyy hh:mm:ss	Sampled data time. CST or SDST depending on the date
Surface Condition	Text	See table 11
Surface Temp	Single	Degrees C
Freeze Temp	Single	Degrees C
Chemical Percent	Integer	Percentage
Depth	Double	Millimeters
Ice Percent	Integer	Percentage
Salinity	Integer	Parts per 100,000
Conductivity	Integer	Mhos

Table 3.5: Data fields of R/WIS sub-surface FIFO file

Data Field	Data Format (ASCII)	Description
Sensor Group ID	12	Sensor group such as R/WIS and Traffic
System ID	Integer	Identifies differences of system
Site ID	Integer	Identifies differences of sites
Sensor ID	Integer	If only one sensor exists, it is 0. Otherwise, uses Mn/DOT number.
Date Time	MM/dd/yyyy hh:mm:ss	Sampled data time. CST or CDST depending on the date
Subsurface Temp	Integer	Degrees C
Subsurface Moisture	Integer	Percentage
Delta-t	Triple	Picoseconds

Table 3.6: Data fields of traffic FIFO File

Data Field	Data Format (ASCII)	Description
Sensor Group ID	20	Sensor group such as R/WIS and Traffic
System ID	Integer	Identifies differences of system
Site ID	Integer	Identifies differences of sites
Sensor ID	Detector ID	
Date Time	Mm/dd/yyyy hh:mm:ss	Sampled data time
Vol	Integer	Count
Occ	Single	0.0 – 100.0 %
Speed	Integer	MPH

3.4. RDBMS Table Design

As indicated before, RDBMS is a convenient tool for medium range data archiving and retrieval usages since it allows complex queries. On the other hand, a care should be taken since RDBMS can add a significant amount of overheads and runs slow when the table size becomes large. A method of coping with a large RDBMS table size is described in Section 3.4.2 in which a partitioning scheme is used.

3.4.1 RDBMS Main-Table Design

The importance of the database design is the maximum flexibility in which data can be searched in any combination of temporal and spatial information. In addition, it must be able to store all types of TSD. The benefit of using RDBMS is maximized by implementing the two mentioned conditions. To that end, the columns of the main data table comprise, Sensor Group ID, System ID, Site ID, Sensor ID, Date and Time, Parameter Name and Parameter Value for each datum. The actual implementation with the column names and specification are summarized in Table 7.

Table 3.7: Main RDBMS Data Table, Table Name = "MainData"

Data Field	Data Format	Description
SenGroup	Integer	R/WIS, traffic as defined
SysID	Integer	Identifies differences of system
SiteID	Integer	Identifies differences of sites
SenID	Integer	
DataTime	Date	Sampled data time
ParName	Text	Parameter type such as AirTemp, vol, ... defined as extension in UTSDF
ParValue	Integer	Defined same as UTSDF Need to check if all values can be expressible with integer.

Along with the main table, multiple parametric definition tables must be created to supplement the description of parameters. Four static supporting tables are defined to support the main table. The first one is the parameter definition table in which the detailed code description on each parameter is stored. The definition of this database table is shown in Table 8. The second, third, and forth supporting tables define the codes for precipitation, precipitation-type, and surface conditions using numeric numbers, which are shown in Table 9 through 11. These tables essentially contain supporting information that is needed to interpret the data in the main table and a common practice in the database design.

Table 3.8: Parameter definition table, Table Name = "ParDef"

Data Field	Data Format	Comment
ParName	Text	Extension used in UTSDF
ParFName	Text	Parameter full name
UnitName	Text	Unit name, such as Percent
Factor	Float	Decimal point position indicator

Table 3.9: Codes for precipitation intensity

Classification	Code
None	0
Light	1
Slight	2
Moderate	3
Heavy	4
Other	5
Unknown	6
Anything else	7

Table 3.10: Codes for precipitation type

Classification	Code
None	0
Yes	1
Rain	2
Snow	3
Mixed	4
Light	5
Light Freezing	6
Freezing Rain	7
Sleet	8
Hail	9
Frozen	A
Unidentified	B
Unknown	C
Other	D
Anything else	E

Table 3.11: Codes for surface condition

Classification	Code
Dry	0
Wet	1
Chemically Wet	2
Snow/Ice Watch	3
Snow/Ice Warning	4
Damp	5
Frost	6
Wet Above Freezing	7
Wet Below Freezing	8
Absorption	9
Absorption at Dewpoint	A
Dew	B
Black Ice Warning	C
Other Slush	D

3.4.2 RDBMS Table Partitioning

Even with the RDBMS table constructed with a relatively small number of data sets as in this prototype, it was noticed that the size of RDBMS table grows very rapidly and the performance of query was significantly degraded. For one year of R/WIS and traffic data, the size of RDBMS grew to 8.7GB, and the time to query a few weeks of data took a significant amount of time (more than one hour). This type of RDBMS problem is well documented in our early study and a good lesson even for the small number sites in TSD [3]. In order to resolve this issue, the main table was partitioned into monthly tables, which resolved the "Session Timeout Expired" error caused by the long retrieval time. Partitioning the main table into monthly tables appears a good solution for R/WIS and traffic data and allowed efficient management of the table size and query time. On the other hand, the penalty paid is increased complexities in query statements when the data is searched for multiple partitions.

3.5. Software Modules

The TSD warehouse in this project consists of data sources, computer hardware, and software modules. During this project, nine software modules were created as the building blocks of the overall system. These software components, database, archive storage, and web server concurrently ran through the networked environment of six PCs. Figure 4 illustrates the relationship between the software components. Five software modules are used to acquire data from the data sources, and to convert the data to a proper format, and to store them at the FIFO storage. Two software modules are used to load the data to RDBMS and archive them into UTSDF storage. Four application software modules utilize the archived data. This sub-section describes the function of each software module.

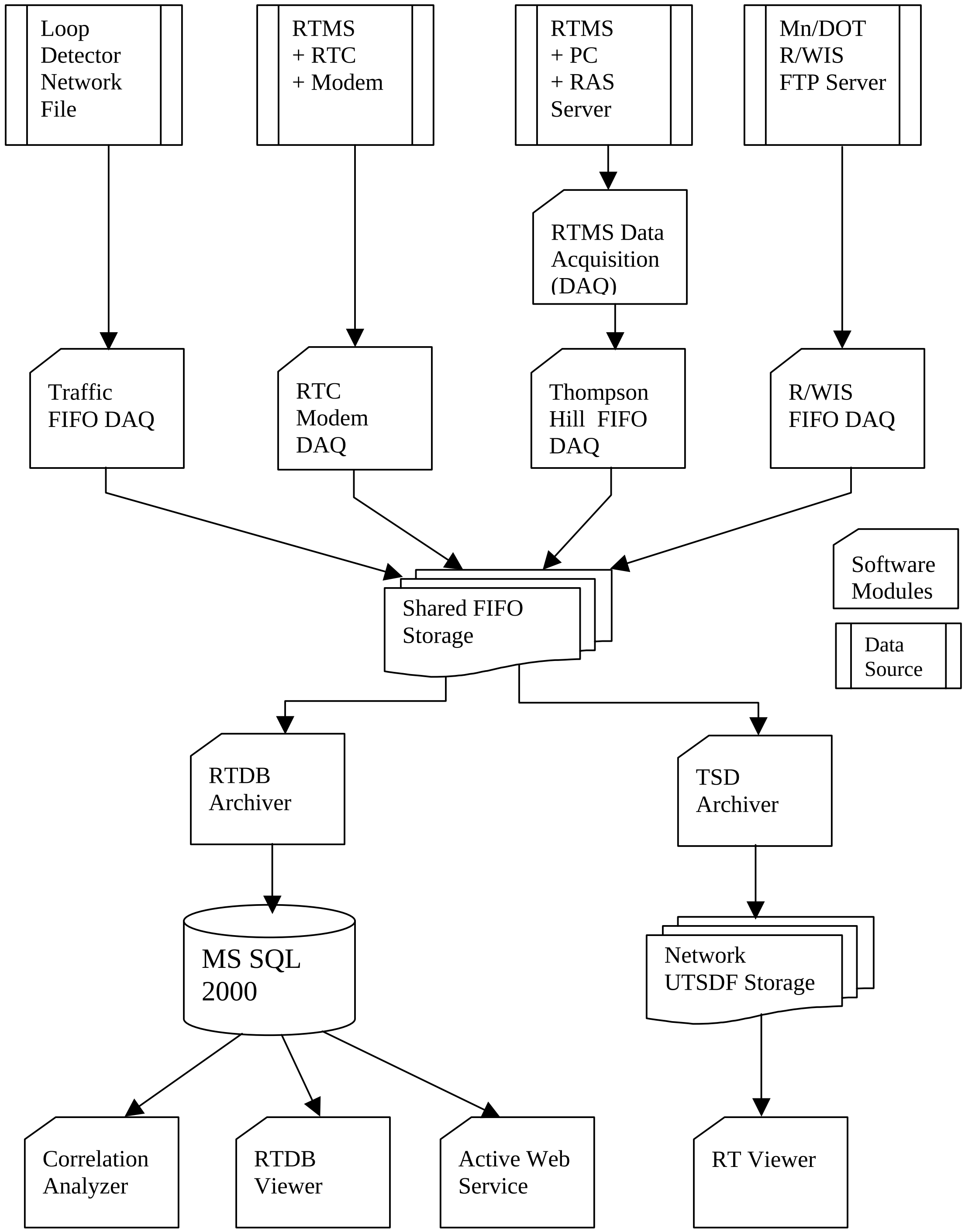


Figure 3.3: Software components

3.5.1 RTMS Data Acquisition Module

This software resides at the Thompson Hill site where a PC is connected to an RTMS through a serial connection. This module allows the PC to directly connect to the RTMS through a stat mode to collect data. The collected data is then stored in a daily file and placed in an export directory of the FTP service implemented on the PC. The PC at the site was set up as a PPP server through a remote dial-up server box. The data collection computer at the TDRL access the field PC through a dialup PPP and uses FTP to download the daily data. It was found that using a remote dial-up server and PPP is faster and more reliable than a common dial-up modem approach since the TCP/IP layer takes care of all communication error conditions.

3.5.2 R/WIS FIFO DAQ Module

This program retrieves the R/WIS data from the Mn/DOT R/WIS FTP site, converts the raw data to the FIFO format, and then stores them in the FIFO directory. For convenience, the timestamps are converted to a local time from GMT. This program was written using a standard FTP protocol.

3.5.3 Thompson Hill FIFO DAQ Module

This program dials up the Thompson Hill RTMS site and establishes a PPP connection from which an Internet connection is made. It then accesses the FTP host and downloads the RTMS traffic data using a standard FTP protocol. The downloaded file is converted to the standard FIFO format and stored into the FIFO directory. Since the underlying protocol for this module is TCP/IP, the same code could be used for other types of communication channels such as wireless and Ethernet.

3.5.4 Traffic FIFO DAQ Module

This program acquires the loop detector data for the sites in the Twin Cities Metro area out of the archived data in TDRL. Traffic data is extracted from the archive file, encoded to a FIFO traffic data format, and stored into the FIFO directory.

3.5.5 RTC Modem DAQ Module

The Garfield site is set up using a data collection module called RTC (RTMS Traffic Counter) that acquires data from the RTMS and stores them into the local memory. One of the serial ports of RTC is connected to a modem. Since the RTC is not an intelligent device, i.e., it does not include error or flow control functions in data communication. A conventional serial modem programming was used in this module. The acquired data is encoded to the FIFO traffic format and stored into the FIFO traffic directory.

3.5.6 RTDB Archiver

This is one of the consuming modules of the FIFO data. It reads data out of the FIFO directory and stores it into the RDBMS. The main database runs on a separate server, so the program connects the database through the TDRL intranet. An ActiveX Data Object (ADO) interface was used for the RDBMS interface.

3.5.7 TSD Archiver

This program archives the data stored in the FIFO directory using UTSDF. It is scheduled to run twice a day, and it archives one day’s worth of data at a time and stores them in a single zip file (i.e., each day has a separate zip file). The data files are deleted from the FIFO directory once they are securely archived. Since UTSDF allows efficient retrieval, any data can be readily searched and retrieved from the archived files.

3.5.8 RT Viewer

This program was written to explore and view the contents of archive files. It retrieves data based on simplified graphical query constructed according to the data structure. It provides both text and graphic retrieval, and renders data in graphs upon the user's request.

3.5.9 RTDB Viewer

This is an application program that queries data from the RDBMS. The query was designed using a graphic interface, and it can render data in text or graphs depending on the user's request.

In addition to the software modules described above, an interactive web server page was developed to provide data through web graphical interfaces. For the development of this software, Microsoft ASP.net and a graphic package was used.

CHAPTER 4

APPLICATIONS

Once the R/WIS and traffic data are integrated into one organized resource (data warehouse) as proposed in this research, applications requiring both types of data can be readily developed. This chapter demonstrates the experiments conducted using the data warehouse created. One of the questions frequently asked by traffic engineers has been how weather factors influence traffic. We will show the results based on the observation of the R/WIS and traffic data retrieved from the data warehouse we constructed.

4.1 Review of Correlation Coefficient

Correlation coefficient indicates the degree of linearity between any two random variables. When sufficient amount of data is available, this measurement provides information on strong, weak, positive, negative, or no correlation information, and frequently used for data analysis. The theory is briefly reviewed [5].

Let X and Y be random variables with joint probability *f(x,y)*. Then the covariance is defined as:

$$\begin{aligned} Cov(X,Y) &= E[(X-\mu_X)(Y-\mu_Y)] \\ &= E[XY]-E[X]E[Y] \end{aligned}$$

where $\mu_X = E[X]$ and $\mu_Y = E[Y]$, i.e. mean or first moment. The correlation coefficient, ρ, between two random variables is then defined as:

$$\rho = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \qquad (1)$$

where σ_X and σ_Y are the standard deviations of X and Y, respectively.

Suppose that we wish to compute the correlation coefficient for n pairs of observations or sampled data, $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Converting Eq. (1) for sampled data gives

$$\rho = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad (2)$$

where $\bar{x}$ and $\bar{y}$ are the averages of $\{x_i\}$ and $\{y_i\}$, respectively. Eq. (2) can be further written as the following form, which is commonly used in actual computation.

$$\rho = \frac{n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \qquad (3)$$

In general, Eq. (3) should be used for the computation of correlation coefficient between any pairs of observed data. If each pair is not observed at the same time, the data should be interpolated to match the sampled time instances. For population correlation coefficient, Eq. (1) should be used.

Correlation coefficient provides useful information in relationship between two variables, which is explained next. The correlation coefficient ρ has the following important properties.

1. The value of ρ does not depend on which of the two variables under study is labeled *x* and which is labeled *y*.
2. The value of ρ is independent of the units in which *x* and *y* are measured.
3. $-1 \le \rho \le 1$
4. $\rho = 1$, if and only if, all (x_i, y_i) pairs lie on a straight line with a positive slope, and $\rho = -1$, if and only if, all (x_i, y_i) pairs lie on a straight line with a negative slope.
5. The square of the sample correlation coefficient gives the value of the coefficient of determination that would result from fitting the simple linear regression model. For example, $\rho^2 = .25$ implies that in a regression of y on *x*, only 25% of *y* variation would be explained by the model.

Rule of thumb in understanding the correlation:

1. The correlation between two variables is **weak** if $0 \le |\rho| \le 0.5$
2. The correlation between two variables is **strong** if $0.8 \le |\rho| \le 1$
3. If $\rho \approx 0$, no apparent relationship exists or it is a nonlinear relationship.

Figure 5 further illustrates the meaning of ρ using scatter graphs.

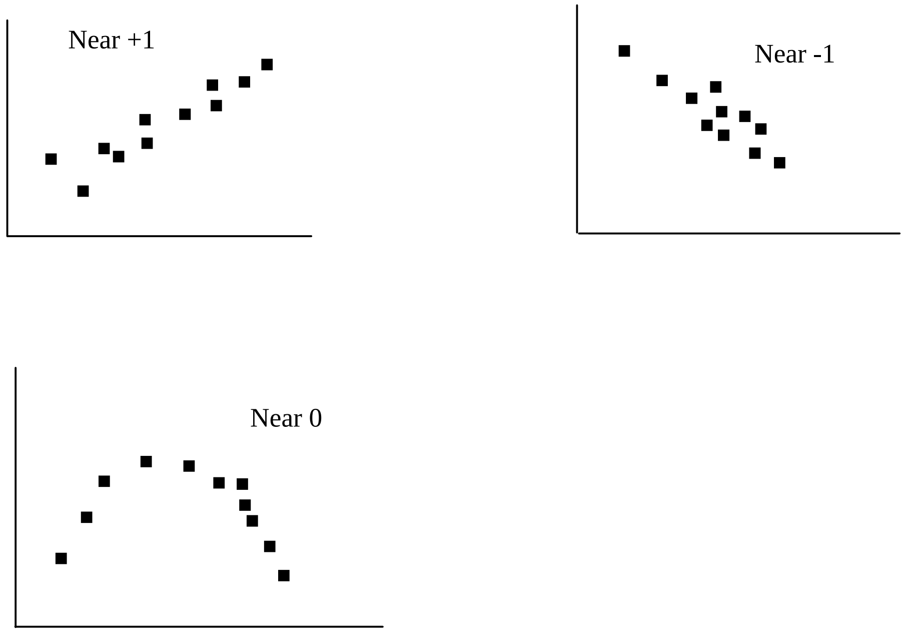


Figure 4.1: Graphical view of correlation coefficients and data trends

4.2 Matrix of Correlation Coefficient

A correlation coefficient matrix is a table that shows the degree of linear correlation between all possible pairs of two parameters within the set of parameters under study. In our case, they are the parameters of R/WIS and traffic from which we wish to identify correlations. The values in the table are correlation coefficients and range from -1 to +1 as described in Section 4.1. The closer it is to either of the extremes the more they are linearly related. For analysis, two months of 2005, January and June, each representing typical winter and summer months were selected and tabulated. Table 12 shows the correlation coefficient matrix computed for January 2005, and Table 13 shows coefficients for June 2005 at the Little Canada site. It can be observed from the Table 12 and 13 that air temperature (atemp) and dew temperature (dtemp) have a strong linear correlation (0.908) so as the wet bulb temperature (wbtemp) and air temperature (atemp). It is also interesting to observe that relhum (relative humidity) has a strong correlation to dtemp (dew temperature). These strong correlations are expected. Correlation between two traffic parameters, occupancy (occ) and volume (vol), is also shown strong, as expected. Wind speed showed almost no correlation to any other weather or traffic parameters. The correlations between traffic and R/WIS parameters were shown to be very weak. For the seasonal differences (difference between Table 12 and 13), summer had somewhat stronger correlations between traffic and R/WIS parameters, but it was not strong enough to say a clear correlation. From this experiment, it was clear that no linear correlation exists between traffic data and atmospheric R/WIS data. Therefore, the correlation data suggests that air temperature, humidity, or dew points have little or no effects on traffic.

Table 4.1: The correlation coefficient matrix for January 2005 at Little Canada

	atemp	dtemp	relhum	avgspd	wbtemp	sur-temp	sub-moist	vol	occ
atemp		0.908	0.418	-0.002	0.995	0.865	0.003	0.028	-0.252
dtemp			0.737	-0.044	0.939	0.662	0.028	-0.087	-0.183
relhum				-0.025	0.462	0.117	0.206	-0.286	0.083
avgspd					-0.019	0.107	-0.174	0.109	-0.207
wbtemp						0.823	0.0599	-0.000	-0.182
sur-temp							0.065	0.136	-0.208
sub-moist								0.197	0.149
vol									0.706

Table 4.2: The correlation coefficient matrix for June 2005 at Little Canada

	atemp	dtemp	relhum	avgspd	wbtemp	sur-temp	sub-moist	vol	occ
atemp		0.387	-0.666	0.338	0.811	0.879	0.353	-0.068	0.038
dtemp			0.470	0.086	0.816	0.085	0.535	-0.340	-0.314
relhum				-0.263	-0.025	-0.787	0.086	-0.226	-0.089
avgspd					0.245	0.380	0.126	0.015	-0.217
wbtemp						0.520	0.406	-0.239	-0.138
sur-temp							0.087	0.098	0.121
sub-most								-0.562	-0.346
vol									0.942

4.3 Effect of Pavement Conditions on Traffic

Another important question that we wish to find out through the data system developed in this research was how pavement surface conditions affect the traffic. To study this effect, one of the well-known trends in traffic data is utilized, i.e., the volume counts of the same weekday at the same location are similar unless the same weekday was one of the holidays or special event days [6]. The idea here is that, for example, the same Tuesdays within consecutive weeks should have similar volume counts unless it is affected by special events such as inclement weather conditions or holidays. Incorporating this idea, traffic volumes were collected for the same weekdays for comparison but by organizing each day with the number of hours in pavement conditions provided by R/WIS data. Table 14 summarizes the retrieved data according to the pavement conditions in number of hours in January 2005 at the Little Canada site. Pavement conditions are classified according to the R/WIS data into five categories, i.e., dry, damp, frost, snow/ice warning, and mixed. Since the month chosen here is January (middle of winter), most of the effects are concentrated in frost and snow warning conditions. A snow condition is recorded when actual snow or ice is detected on the pavement surface from the pavement sensors. From Table 14, the trend can be clearly observed that days involving frost and snow conditions reduce the total daily traffic volume. The first week of January does not follow the effect of pavement conditions since it is a holiday week, but a reduction of traffic volume is clearly observed when the snow and frost conditions are present in the second and third week. Additional data is shown in Appendix 5, and a similar effect is observed. In any case, it is interesting to notice that while air temperature and humidity have little or no effects on traffic, pavement conditions clearly affect traffic volume. Unfortunately, pavement conditions are non-numeric parameters, and direct correlation coefficients can not be computed. Table 14 includes the total traffic volume of a day when snow or frost events occurred, but it does not give a direct comparison of volume changes at the time of actual snow/ice events. In the next section, we further analyze the data through color coded line graphs to understand how much the snow/ice pavement conditions affect the traffic volume at the time of the events.

Table 4.3: Weekly Surface Condition and Traffic Volume at Little Canada in January 2005

			Surface Conditions in number of hours				
Day	**Weekday**	**Traffic Volume**	**Dry**	**Wet**	**Snow**	**Frost**	**Damp**
3	Monday	105494	24	0	0	0	0
10	Monday	111185	9.17	0	14.83	0	0
17	Monday	101698	13.17	0	10.83	0	0
24	Monday	109490	5	0	10.83	0	8.17
31	Monday	108200	24	0	0	0	0
4	Tuesday	111863	24	0	0	0	0
11	Tuesday	112362	4	0	20	0	0
18	Tuesday	107450	13.5	0	9.5	1	0
25	Tuesday	112567	6	0	8.83	0	9.17
5	Wednesday	113963	24	0	0	0	0
12	Wednesday	107765	0	0	15.5	4	4.5
19	Wednesday	113696	12.5	0	11.5	0	0
26	Wednesday	114215	17.83	0	3.83	0	2.33
6	Thursday	115646	24	0	0	0	0
13	Thursday	113157	17.83	0	6.17	0	0
20	Thursday	110750	0	0	24	0	0
27	Thursday	116972	23.83	0	0.17	0	0
7	Friday	118569	0.33	0	23.67	0	0
14	Friday	114673	24	0	0	0	0
21	Friday	91260	0	0	24	0	0
28	Friday	122897	18.17	0	5.83	0	0
8	Saturday	88326	12.33	0	11.67	0	0
15	Saturday	80185	24	0	0	0	0
22	Saturday	73764	9.17	0	14.5	0	0.33
29	Saturday	73764	24	0	0	0	0
2	Sunday	66020	12.33	0	9.33	2.33	0
9	Sunday	71561	0	0	23.83	0.17	0
16	Sunday	66464	8.17	0	15.83	0	0
23	Sunday	71857	14.5	0	9.5	0	0
30	Sunday	77639	24	0	0	0	0

4.4 Effect of Traffic Volume on Snow, Ice, and Frost Conditions

Although Table 14 allows us to observe the total volume trends, it does not show when and how much the pavement conditions affected the volume. In order to see the effect based on the time of the day, two same weekdays in different weeks were plotted side-by-side with comparable dry days. Figure 6 shows volume changes of two dry pavement conditions of Tuesdays in February 2005 (the data was collected from the Little Canada site). Typical daily traffic volume at this site is 111,000 to 123,000. It can be noticed that volume trends and the quantities of two dry days are similar showing similar morning and afternoon peak levels of volume. Therefore, it suggests that the dry condition of the week of the days could be used as a baseline for traffic volume effects by pavement conditions.

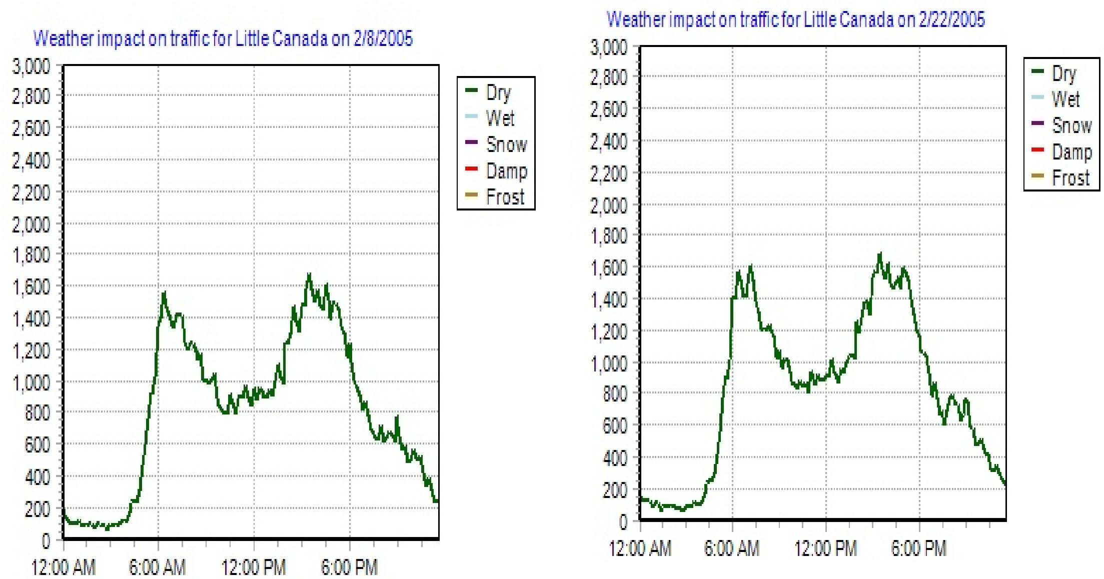


Figure 4.2:Volume vs. hourly surface conditions at Little Canada on Tuesdays 2/8/2005 and 2/22/2005.

In order to analyze the effect of snow/ice and frost conditions on traffic volume, days involving such weather conditions were plotted on a volume line using a color code for different surface conditions. The plots for the comparable traffic conditions (i.e., same day of week compared with dry days) were shown in Figures 7-9. The colors were coded as green for dry, light blue for wet, purple for snow/ice warning, red for damp, and dark yellow for frost conditions. From these figures, it can be clearly observed that the volumes slightly decreased when the day had snow/ice events. For example, Figure 7 shows that the morning peak (6:00-9:00AM) volume is 24,563 on a dry condition and 21,194 during snow events, decreasing about 15% during the snow events. The decrease of the volume is less significant in Figure 8., which indicates that the snow/ice conditions on that day were less severe. This is observable from the graph that the day includes dry pavement conditions between 12:00 to 5:00 p.m. In this case, the morning peak volume was 23,998 on Jan 4th (dry condition) and 23,462 on Jan 11th (snow condition). The effect on traffic on the same days (Jan 4th and 11th) is shown in Figure 10. Notice that if the traffic level is similar more congestion occurs with snow conditions, which indicates reduction of capacity of the road. Figure 9 shows another case. Again, snow/ice conditions were observed during the morning. The morning peak volume was 18,596 on Jan 17th (snow) and 21,601 on Jan 31st (dry), decrease of 15%. Although no snow events occurred during the afternoon, it is interesting to observe that the volume is still low. This indicates that people traveled less due to the morning snow event. We may deduce from these observations that the amount of traffic volume decrease depends on the severity of the snow/ice conditions rather than the total number of hours in that conditions. From these experiments, we may draw an analysis that inclement weather conditions discourage people from driving and, consequently, the volume count is reduced. When the volume level is not reduced even if the snow events occurred, more congestion is observed.

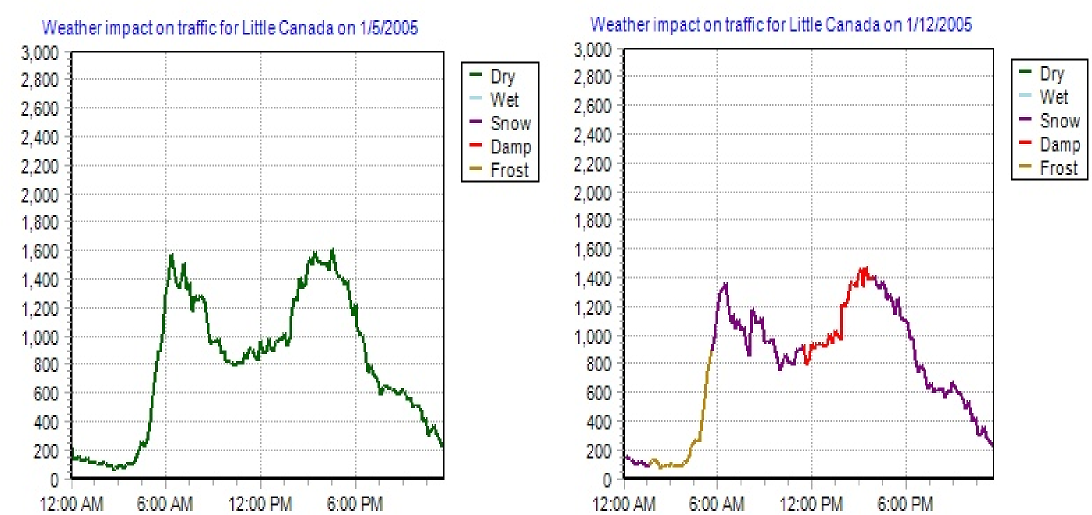


Figure 4.3: Volume vs. hourly surface conditions at the Little Canada station on Wednesdays, 1/5/2005 and 1/12/2005.

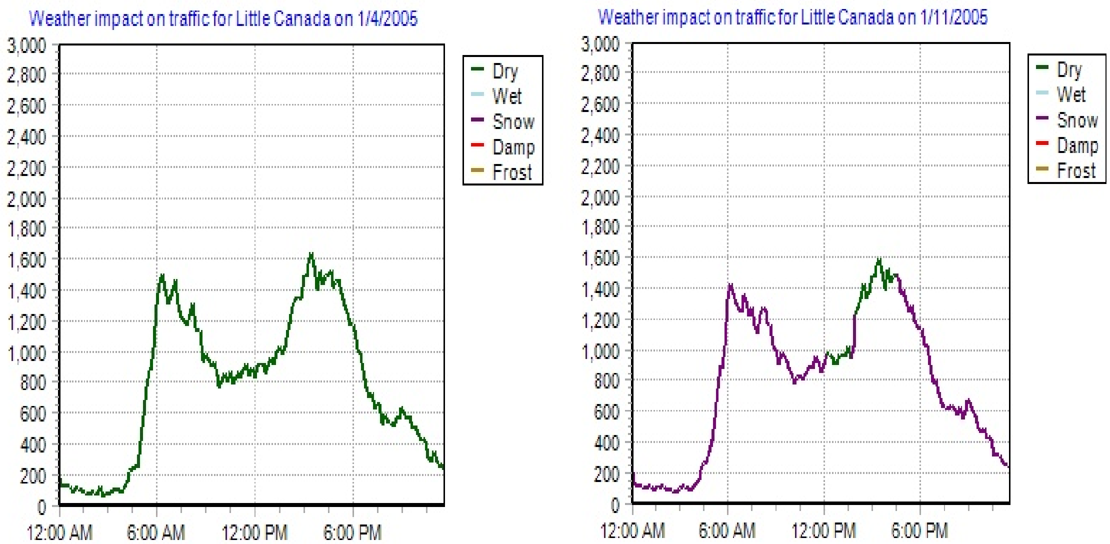


Figure 4.4: Volume vs. hourly surface conditions at the Little Canada station on Tuesdays, 1/4/2005 and 1/11/2005.

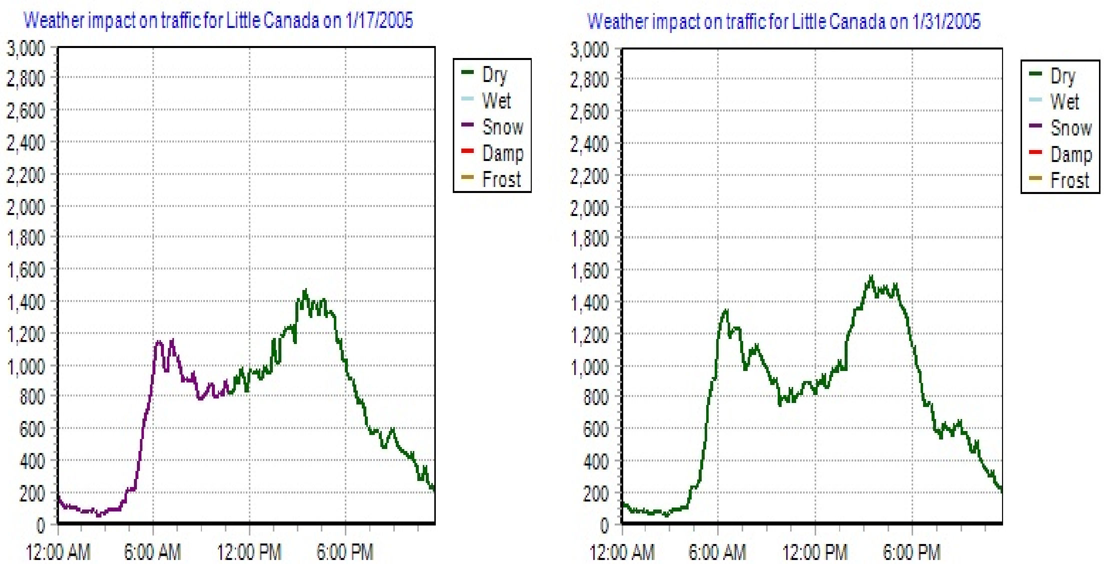


Figure 4.5: Volume vs. hourly surface conditions at the Little Canada station on Thursdays, 1/17/2005 and 1/31/2005.

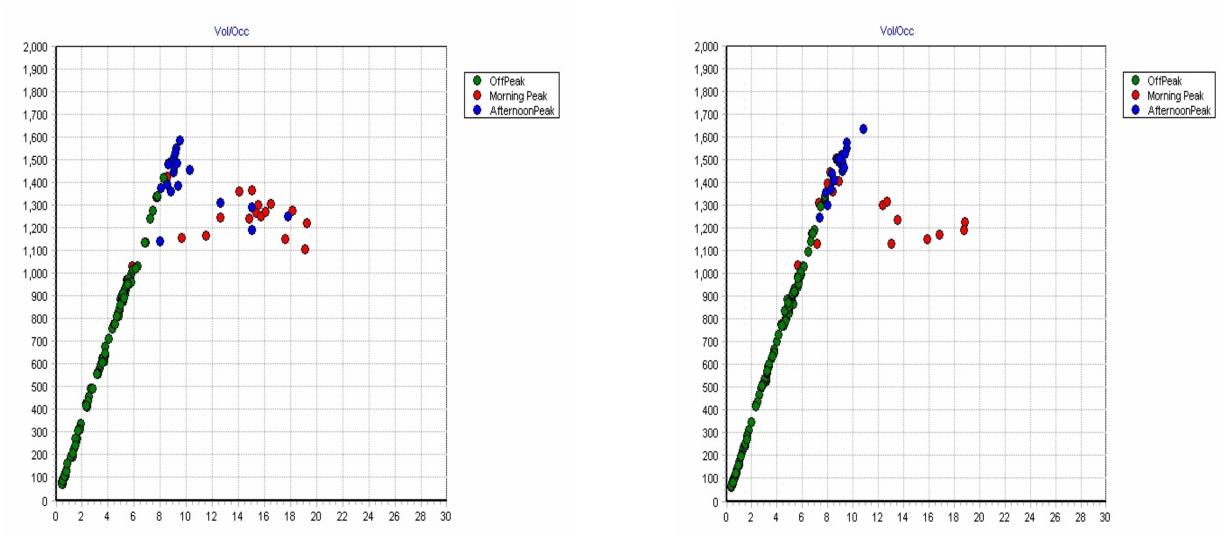


Figure 4.6: Volume vs. occupancy relation at the Little Canada station on Tuesdays, 1/4/2005 and 1/11/2005.

4.5 Effect of Traffic Volume on Wet Conditions

It is also interesting to learn how wet pavement conditions affect traffic volume in summer months. For that we repeat the same plotting scheme. Figure 10 shows two consecutive Wednesdays with damp conditions reported by R/WIS. Notice from the morning peaks that May 15^{th} had more damp conditions than May 25^{th}, and a slight drop (7%) in traffic volume can be observed. The actual morning peak volume was 22,455 on May 18^{th} and 24,250 on May 25^{th}.

Figure 4.7: Volume vs. hourly surface conditions at the Little Canada station on Wednesdays, 5/18/2005 and 5/25/2005.

Another example is shown in Figure 11, taken from the Maple Grove station for two Thursdays. June 16^{th} was a dry day and shown as a baseline, and May 18^{th} had wet conditions during most of the day. Since R/WIS reports wet when grounds are saturated with water, it was clearly a rainy day on May 18^{th}. The morning peak volume was 24,280 on June 16^{th} and 22,148 on May $18^{th.}$ In this case, traffic volume dropped about 9% due to the rain condition. It was an interesting result that rains in Minnesota can affect traffic that dramatically. The data suggests that heavy rain events discourage people from driving.

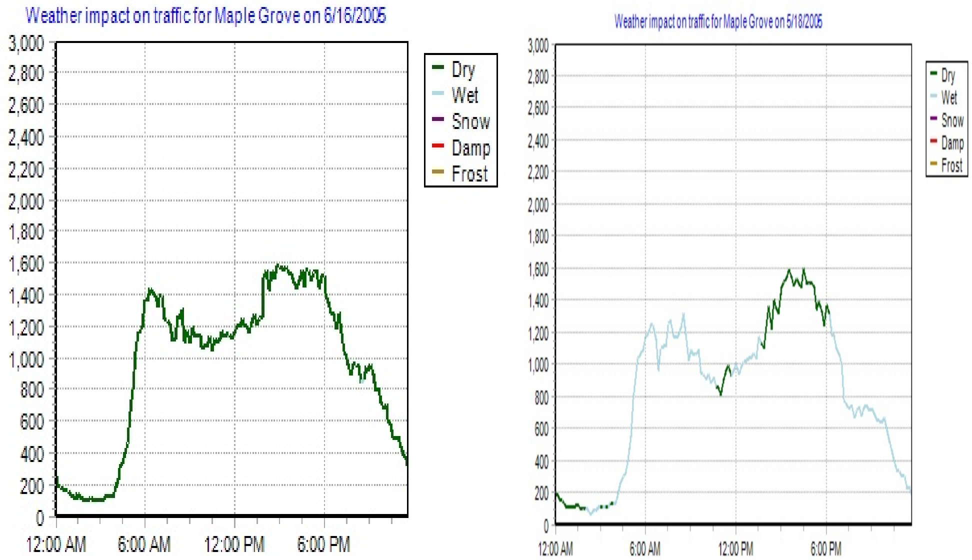


Figure 4.8: Volume vs. hourly surface conditions at the Maple Grove station on Wednesdays, 6/16/2005 and 5/25/2005.

CHAPTER 5

CONCLUSION

In the past, R/WIS and traffic data have mostly been managed in isolation, and thus the benefits attainable by correlating both types of data have been underutilized. Researchers believe that historical and real-time traffic data along with R/WIS data can lead to better information. The focus of this project was to develop an efficient data integration model for R/WIS and traffic data. The basic data architecture was developed from the ground up starting from the sensor data and building a data warehouse for retrieval applications. For testing the model, RTMS and loop detectors were used for traffic data, and the Mn/DOT's statewide R/WIS was used for the R/WIS data. The final prototype developed was a functional data warehouse that provides efficient retrieval and archiving of both traffic and R/WIS data. The prototype consists of nine independent software modules that interact between the data warehousing building blocks through on-line communication and automation. The prototype also includes a standard SQL database and archived network storages that allow multi-user on-line accesses. A new concept of network FIFO storage was developed and implemented, and was the key in which loss of data by malfunctioning sensors or communication was minimized. The prototype implemented worked well according to the design objectives and demonstrated its self-maintainability without any user intervention. All available sites where both types of data exist were integrated to the prototype developed in this project. Although the prototype system was designed for only eight sites, this system is capable to scale up to a large data warehouse that could hold traffic and R/WIS data for single or multiple states.

After completion of the prototype data warehouse, several examples of retrieval applications were developed to test the functionality, to graphically view data, to compute correlations, etc. In all cases, the system allowed retrieval of any combination of R/WIS and traffic data in real-time without much difficulty. As a practical application of this highly integrated data warehouse, this research included a study on how inclement weather conditions affect traffic flow. For retrieving data for this study, traffic volume data was retrieved according to pavement conditions, and tabulated to demonstrate the benefits of data integration. From this table, a clear trend on drop of traffic volumes on the days with special weather events, i.e., damp, frost, snow, and ice, was observed. Next, hourly traffic volume counts were plotted for the same weekdays in different weeks using color codes that represent pavement conditions. It was clearly observed that traffic volumes were indeed reduced depending on the severity of snow/ice conditions. It was also found that if a part of the day includes snow/ice on the pavement conditions, the overall traffic volume was decreased even for the dry portion of the day.

Integrating traffic data with R/WIS data has not been a common practice in transportation agencies including Mn/DOT. Nevertheless, it was shown through this project that a well-designed data warehouse and data models can be readily built, and many benefits from integrated treatments of different types of data are attainable in addition to the benefits of the traditional independent treatments of R/WIS and traffic data. It was also shown that the data warehouse does not need to be complicated nor expensive. The data warehouse model suggested in this research is an inexpensive solution, and yet it would be sufficient to scale up to a statewide level traffic and R/WIS data system.

REFERENCES

[1] U.S. DOT ITS, Archived Data User Service (ADUS), "ITS Data Archiving: Five Year Program Description," (Published by U.S. DOT, ADUS Program, March 2000).

[2] T. M. Kwon and Ed Fleege, "R/WIS Architecture for Integration and Expansion," *Journal of the Transportation Research Board*: *Transportation Research Record 1700*, pp. 1-4, The National Research Council, (The National Academies, 2000).

[3] T. M. Kwon, N. Dhruv, S. Patwardhan, E. Kwon "Common Data Format Archiving of Large-Scale Intelligent Transportation Systems Data for Efficient Storage, Retrieval, and Portability," *Journal of the Transportation Research Board*: *Transportation Research Record 1836*, pp. 111-117, (National Academy of Science, 2003).

[4] T.M. Kwon and N. Dhruv, "Unified Transportation Sensor Data Format (UTSDF) for Efficient Archiving and Sharing of Statewide Transportation Sensor Data," *Proc. of the Transportation Research Board 83nd Annual Meeting*, Washington D.C., (Jan. 2004).

[5] J. L Devore, *Probability and Statistics for Engineering and the Sciences*, 4th Ed, (Brooks/Cole Publishing Company, CA, USA, 1995).

[6]T.M. Kwon, *TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program*, Minnesota Department of Transportation, Report No. MN-RC-02004-29, (July, 2004).

APPENDIX A

RTMS SETUP AND COMMUNICATION DESIGN

A.1. RTMS Remote Access Systems: Configurations

Three Remote Traffic Microwave Sensor (RTMS) sensors were installed in District-1 for monitoring traffic. Two different types of system configurations were implemented as listed in Table A.1, and each of them is further described along with the experiences learned.

Table A.1: RTMS Configuration

Location	System Configuration
Garfield Av.	RTMS+RTC+Modem
Atkinson Bridge	RTMS+RTC+Modem
Thompson Hill	RTMS+PC+RAS Server

A.1.1 RTMS + RTC + Modem Configuration

RTMS is a microwave-based radar traffic sensor, which provides the traffic presence information. However, RTMS itself does not have the required storage for interval counting. In order to record vehicle volumes or occupancy, another module with a data storage is needed. **R**TMS **T**raffic **C**ounter (RTC) is a microcomputer with non-volatile memory that communicates with RTMS and saves the data received from RTMS. RTC is separately sold by the EIS company.

Figure A.1 shows the connection diagram of the RTMS+RTC+Modem configuration. For RTMS testing, the male end of a regular DB9 modem cable/connector can be connected to the DB9 female end from RTMS, and the other end to a serial port of a PC. The RTMS software on the PC then should be able to directly read the data from RTMS. Similarly, a regular DB9 modem cable/connector can be connected to the female DB9 port of the RTC for directly readingdata from RTC using a PC. A typical analog modem (POT modem) can be used, but the transmission speed is limited to 9600 baud rate by the RTC and RTMS hardware setup.

The RTMS+RTC+Modem configuration is recommended by the manufacturer. However, the experience of this configuration through this project was very negative, and not recommended for future implementations due to the following reasons.

- Due to the 9600 baud rate limitation, data transmission is slow.
- RTC does not provide any protocol for reliable data transmission such as acknowledgements.
- RTC frequently does not respond to a proper command, and there is no easy way of verifying whether the command was received or not.
- The real time data during the data transmission is lost.
- It is not reliable to collect real time data (due to frequent data losses).

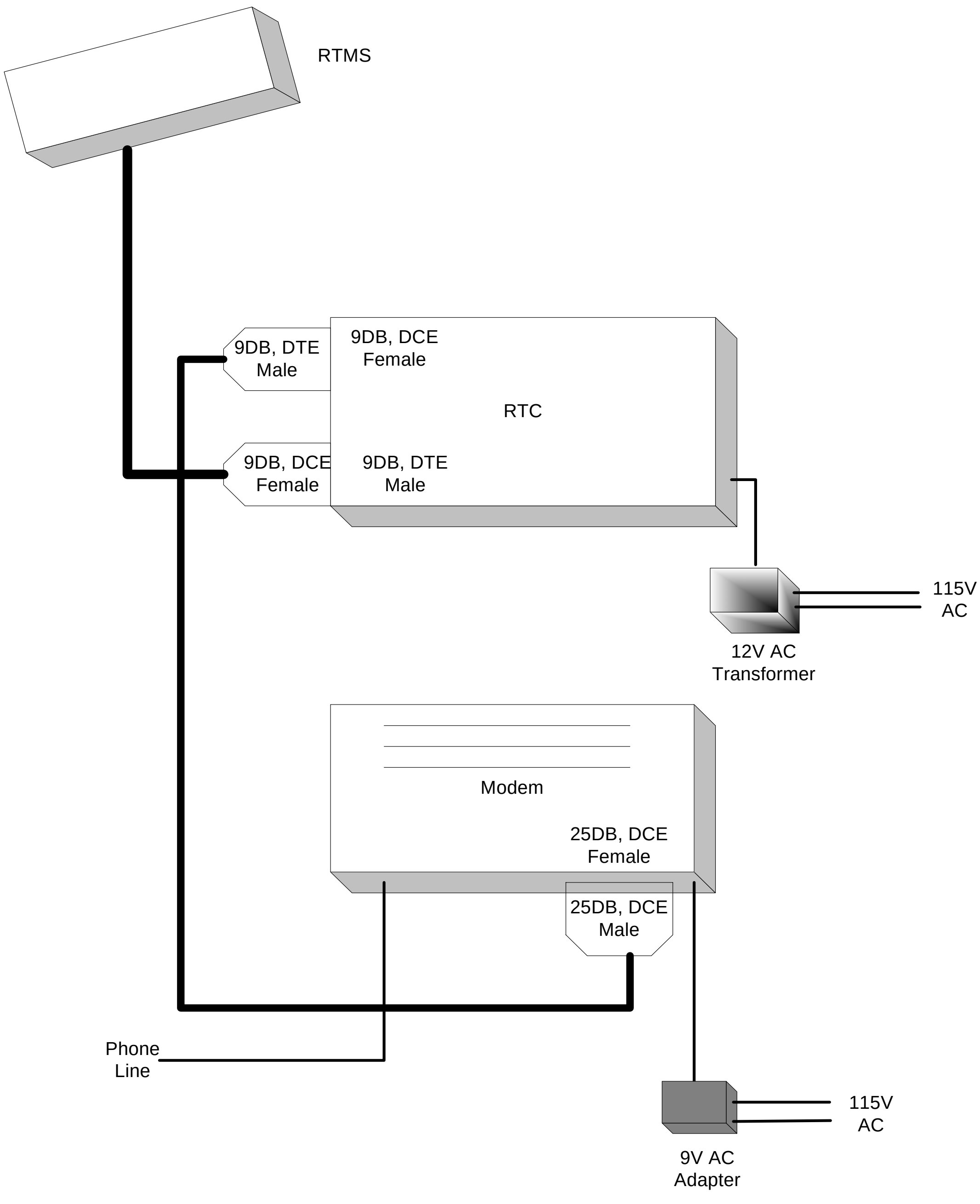


Figure A.1: RMS+RTC+Modem configuration.

The modem connected to an RTC must be set up correctly in order to avoid frequent lockups. The lockups appear as a continuous busy signal, and only a manual power reset can unlock the modem. A technique that can be used to avoid the lockups is to set up the modem in such a way that it hangs up if no activities are observed on the line for an extended period of time (i.e., the modem can be set up to hang up after 3 to 5 minutes). The modem used in this project was a US Robotics Sportster 33.6 and was set up for automated reply for one ring using 3 wire connections (Tx, Rx, G) with 9600 baud rate. For this setting, the following AT commands must be manually entered using a terminal interface to change the factory default setting to the desired auto-hangup setting.

AT&H0	Flow control disabled (0 = zero)
AT&R1	Modem ignores RTS
ATS19=5	Hang up if no data activity is observed for 5 minutes
AT&N6	Set 9600 as the ceiling baud rate (allow step down rate if line is noisy)
AT&w0	Write current configuration to NVRAM0 (Nonvolatile RAM zero, Y0)
ATY0	Set profile 0 as the default, i.e., profile 0 is loaded for power-on reset

The eight toggle switches must be set as follows.

Switch Number	Switch Position	Meaning
1	Down	DTR override
2	Up	Verbal result code
3	Up	Suppress result code
4	Up	Echo off-line commands
5	Up	Auto answer on the first ring
6	Up	Carrier detect normal
7	Up	Load NVRM defaults
8	Down	Smart mode

Most analog modems have similar types of AT commands and settings that can be configured as the above example.

A.1.2 RTMS + PC + RAS Server Configuration

According to the experience from this project, a reliable remote RTMS data collection system can be designed using a PC and a RAS server as shown in Fig A.2. In this configuration, a PC or PC104 directly connects to the RTMS serial port and collects data using a STAT mode (one of the RTMS mode) and saves the data to a PC FTP site. This FTP site is then accessed through a Remote Access Service (RAS) server in which the PC side is connected through an Ethernet port. A typical RAS server includes several built-in analog modems in the box. Therefore, an analog phone line is directly connected to the RAS server. For the RAS server, this project used the RAS Finder developed by the MultiTech Systems. This RAS server contains three V.90 modems which would allow connections up to three phone lines. In this project, only one of the modems was used, and the other two were not used.

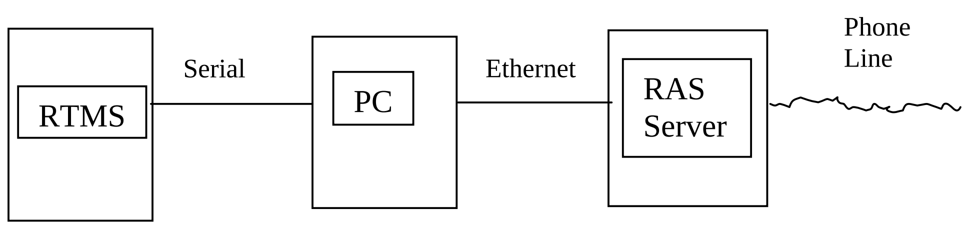


Figure A.2 RTMS + PC + RAS Server Configuration

The main advantage of this configuration is that the reliability provided by TCP/IP is fully utilized and the full speed of the modem can be utilized. In addition, this configuration does not require any setup concerns about the modem, since the RAS server takes care of its internal modem. Software development is also much simpler than the serial port downloads of data through a modem. In addition, regular FTP protocol commands can be used to automate the data collection process. However, a drawback of this configuration is the requirement of a PC that can be very bulky. This can be readily solved by replacing the PC with a single board PC104 or EPIC systems. By doing so, it also increases the reliability of the overall system, since PC104 or EPIC cards do not use hard disks.

A.1.3 RTMS Sensor Installation Guide

The RTMS microwave beam and its detection zones are specified as follows:

Elevation beam-width:	45 degrees
Azimuth beam-width:	15 degrees
Range:	3 to 60 meters (10 to 200 ft)
Number of detection zones:	up to 8 zones

Using the above information, a RTMS installation guideline was developed and given below. This guideline should be given to the installer.

- Minimum distance from the first detection zone should be 3 meters (10ft). If more than three lanes are monitored, the minimum should be increased to 5 meters (17ft).
- The height of the RTMS sensor unit should be 5 meters from the road surface if the first detection zone is less than 5 meters from the mounting pole.
- If the first detection zone is 5-9 meters (17-30ft) away from the mounting pole, the height should be adjusted to 5-9 meters (17-30ft) from the road surface.
- For 1 to 4 lanes, the RTMS head should point to the middle of the lanes to be monitored.
- For 5 to 8 lanes, the RTMS head should point to the first 1/3 of the entire detection zone.

A.2 RTMS Programming

The RTMS programming is done through exchange of messages using a standard serial port. The baud rate is fixed at 9600. The RTMS protocol consists of messages generated by the RTMS and requests received by the RTMS. All messages include the following fields.

Header	Qualifier	Length	Data1	Data2	-------	DataN	Checksum

- The Header is always hex FF
- The Qualifier defines the nature of the data.
- The Length defines the number of bytes in the data field
- The checksum is calculated using the data bytes only. It is modulo 8-bit addition of (Data1 + Data2 + … + DataN), i.e., 8 LSBs of the sum.

The RTMS receives the following messages

Name	Qualifier (hex)	Length	Content of data bytes
BIT Req.	88	0	Built-in test request in Normal mode
P.BIT	90	1	Built-in Test request in polled mode. Data contains the Station ID
Data Req.	8F	1	Data request – polled mode only. Data contains Station ID
Buffer Flush	1F	1	This message is generated on successful receipt of data and clears buffer of the RTMS identified by Station ID

The RTMS statistical data is transmitted at the end of the message period or in response to a poll (Data Request message) as a block of 4 messages. Each of the messages of the block contains per detection zone information and additional information pertaining to all messages in the block (such as message number and sensor ID):

Name	Qualifie (hex)	Length	Content of data bytes
Volume	10	9	First 8 bytes contain volume for each of the 8 detection zones. Undefined zones contain Hex F0 The 9th byte contains the message number
Occupancy	11	9	First 8 bytes contain occupancy for each of the 8

Name	Qualifier (hex)	Length	Content of data bytes
			detection zones. The 9th byte includes sensor ID
Speed	12	11	First 8 bytes contain speed for each of the 8 detection zones (in kph). Hex F0 is placed in bytes corresponding to zones which are not selected The next two bytes hold the direction and average speed of the forward-looking mode. The direction byte will contain 0 or Hex 80 (0 or 1 in most significant bits; all other bits are 0). The average speed byte will contain Hex F0 and the direction byte will contain 0 if the unit is in side-fired mode For RTMS in the forward looking mode and equipped with Rev. 5.0 software the first 8 bytes of the SPEED message contain the volume of vehicles in each of the 8 speed bins. The sum of all speed bins should be equal to VOLUME of zone 2 The 11th byte contains status information. The 4 MSB bits show a temperature calibrator and should be between 1 and 7. 0 is a fault. The LSB bits are usually 0. If 1 appears, do a self test several times
Vol Long	1B	9	First 8 bytes contain volume of long vehicles for each of the 8 detection zones in a side-fired mode. For RTMS in the forward looking mode and equipped with Rev. 5.0 software the VOLLONG message 8 bytes contains the volume in the length bins defined on set-up Byte 9 is spare for future use.

Other messages, generated by the RTMS are delivered at or near real time, provide the following

Name	Qualifier (hex)	Length	Content of data bytes
Targets	18	4	This message is delivered every 100ms in Normal data mode. It provides presence in 32 detection slices and is used by a set-up PC to show targets against the distance scale. Targets are used in the set-up process to position the detection zones
RTMS stat	19	11	This message, delivered every 5 seconds in Normal mode only, provides technical diagnostic information

Name	Qualifier (hex)	Length	Content of data bytes
			for use by EIS staff.
Diagnostic	1D	3	This message provides results of a built-in test request. Each bit in the first byte identifies a mode of failure. Second and third byte identify the RTMS software release version. The failure modes identified by BIT are: bit1 (LSB) Power Supply Fault bit 2 Modulator Signal Fault bit 3 Microwave Module Fault bit 4 Temperature Calibrator Fault bit 5 Modulator Memory Fault bit 6 DSP Fault bit 7 Program Memory Fault bit 8 (MSB) ADC Fault

For multi-drop configuration, a different set of qualifiers are used, which are listed below. Address 0xFF means ‘broadcast mode’ e.g. all nodes are responding to request.

DATAREQ – qualifier – 0x8F Data request (statistical or self test results)
| Header | Qualifier | Length | Address | ChkSum |
0xFF 0x8F 0x01 0x00 – 0xFF

BITPOLLREQ – qualifier – 0x90 Self test in multi-drop configuration.
| Header | Qualifier | Length | Address | ChkSum |
0xFF 0x90 0x01 0x00 – 0xFF

FLUSHBUFREQ – qualifier – 0x1F Remove data from buffer
| Header | Qualifier | Length | Address | ChkSum |
0xFF 0x1F 0x01 0x00 – 0xFF

<u>Respond from RTMS for self-test (BITPOLLREQ)</u>

SPBIT – qualifier – 0x1d – results of self-test in multi-drop config.
| Header | Qualifier | Length | Address | BIT | ProgID | ProgRev | ChkSum
0xFF 0x90 0x04 0x00 – 0xFF *

* BIT – self-test result byte
D0 - Power supply fault (Lsb)
D1 - Modulator fault
D2 - Microwave module fault
D3 - Temperature calibrator fault
D4 - Modulator memory fault
D5 - DSP fault
D6 - Program memory fault
D7 - Gain too low or ADC fault (Msb)

A.3 RTC Programming

For writing a program for modem access to RTC, a different set of commands is used. The information below is needed for system integrators and software developers.

RTC MESSAGES FORMAT

Each byte is denoted by |##| notation below, where ## is a byte of data.

Item	Direction	Description	Format
1.	PC->RTC	Read Clock request	|FF|24|00|00|
2.	PC->RTC	Clear memory	|FF|27|00|00|
3.	PC->RTC	Read memory address	|FF|29|00|00|
4.	PC->RTC	Self test request	|FF|28|00|00|
5.	PC->RTC	Download data req.	|FF|21|00|00|
6.	RTC->PC	Clock data |FF|2F|08|Sec|Min|Hrs|Day|Dte|Mth|Yer|Flg|CheckSum|	
7.	RTC->PC	Self Test results	|FF|30|03|Res|Cod|Rev|CheckSum|
8.	RTC->PC	Memory address	|FF|31|03|Lsb|Msb|Hsb|CheckSum|
9.	RTC->PC	Date	|FF|16|07|Sec|Min|Hrs|Day|Dte|Mth|Yer|CheckSum|
10.	RTC->PC	Time	|FF|17|03|Sec|Min|Year|CheckSum|

Few notes on fine points for developers:

- The checksum is calculated using the data bytes only. It is a modulo 8-bit addition of (Data1 + Data2 + … + DataN).
- When you download, the order of data frames you receive is Date, Speed, Long Volume, Volume, and Occupancy.
- When you issue the “Clear memory” command, the new address starts from 10 (not 0).

- After the “Clear memory” command, the message number does not start from 0. It continues from the last message number.

As an example, a response from RTC for “Read Clock request” is described. The “Read Clock request” command string is formatted as a hex string |FF|24|00|00|. Upon sending a “Read Clock request” string to RTC, RTC returns the following data as an example.

FF, 2F,08,36,44,15,01,23,03,00,FF,B5

This data stream can be interpreted as follows.

FF	2F	08	36	44	15	01	23	03	00	FF	B5
Header	Read Clock	Length	Sec	Min	Hrs	Day	Dte	Mo	Yr	Flag	Cksum

Note that the clock data is expressed in BCD. So the date and time should read as “15:44:36 03/23/00”

The checksum is obtained by taking 8 least significant bits of the hex addition, i.e., (36+44+15+01+23+03+00+FF=B5).

A.4. RTMS Statistics Only Mode

The RTMS Statistics Only Mode is ideal for data collection applications and selected from the RTMS Data menu by selecting “Stat.” In this mode, RTMS reports traffic data at the end of each message period (1 minute in our case). Unlike a normal mode, this mode does not transmit the target blips every 100ms, such that the valuable CPU time can be freed and the data collection task can be run at the background without much effect on other computational operations.

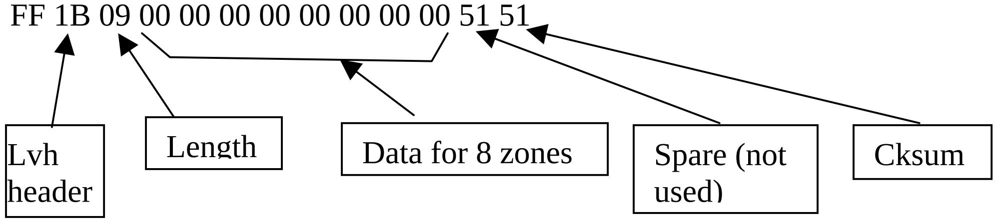
In the Statistics Only Mode, RTMS sends four frames of data in the following order at the end of every message period.

- Long Vehicle Count
- Volume Count
- Occupancy
- Speed

The detailed information on each frame is described using examples below:

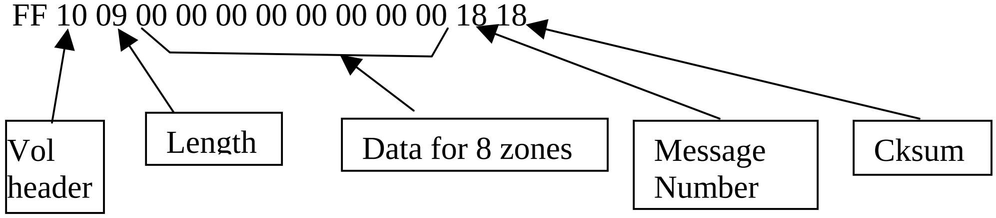
<u>Long Vehicle Count</u>

FF 1B 09 00 00 00 00 00 00 00 00 51 51



<u>Volume Count</u>

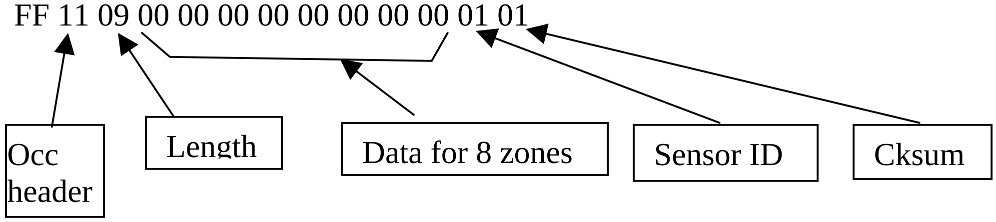
FF 10 09 00 00 00 00 00 00 00 00 18 18



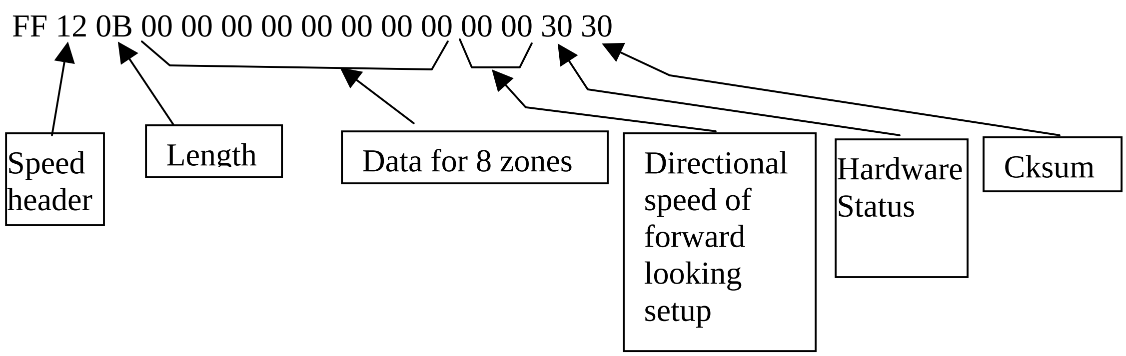
Note: Message number is sequentially increased with mod ff, i.e., 0, 1, 2, ... ff, 0, 1, 2, ...

<u>Occupancy</u>

FF 11 09 00 00 00 00 00 00 00 00 01 01



Speed



Therefore 54 total (13*3 + 15) bytes are transmitted every message period. The information provided here is sufficient to write a program to capture the RTMS interval data.

APPENDIX B

TABLES SHOWING EFFECT OF PAVEMENT CONDITIONS ON TRAFFIC VOLUME

In the following two tables, daily traffic volumes are compared among the same day of the weeks at the Little Canada site. Number of hours in different pavement conditions is specified for each day.

			Surface Conditions in number of hours At Little Canada, Feb 2005				
Day	**Weekday**	**Volume**	**Dry**	**Wet**	**Snow**	**Frost**	**Damp**
1	Tuesday	115129	17.17	0	2.17	0	4.67
8	Tuesday	116006	24	0	0	0	0
15	Tuesday	116851	20.33	0	3.67	0	0
22	Tuesday	117242	24	0	0	0	0
2	Wednesday	116702	7.50	0	6.50	3	7
9	Wednesday	115167	10.83	0	13.17	0	0
16	Wednesday	118256	19	0	5	0	0
23	Wednesday	119273	16.83	0	7.17	0	0
3	Thursday	117377	11	0	6.50	1.67	4.83
10	Thursday	118804	13.67	0	10.33	0	0
17	Thursday	119314	22.33	0	1.50	0	0
24	Thursday	116874	3.67	0	16.33	0	4
4	Friday	123170	10	0	0	0	14
11	Friday	124897	11.17	0	4.50	5.17	3.17
18	Friday	123273	21.83	0	2.17	0	0
25	Friday	124578	11.50	0	12.33	0	0
7	Monday	111779	24	0	0	0	0
14	Monday	109816	10.83	0	1.50	5.33	6.33
21	Monday	108469	23.67	0	0.33	0	0
28	Monday	113646	13.83	0	10.17	0	0

			Surface Conditions in number of hours At Little Canada, Mar 2005				
Day	**Weekday**	**Volume**	**Dry**	**Wet**	**Snow**	**Frost**	**Damp**
1	Tuesday	118043	18.83	0	5.17	0	0
8	Tuesday	117579	23.67	0	0	0	0.33
15	Tuesday	118182	15.83	0	8.17	0	0
22	Tuesday	118522	22	0	1.83	0	0.17
29	Tuesday	120164	24	0	0	0	0
2	Wednesday	120475	21.33	0	2.67	0	0
9	Wednesday	118606	22.67	0	0.33	0	1
16	Wednesday	118769	13	0	7.33	0	0
23	Wednesday	119796	23	0	0.83	0	0.17
30	Wednesday	116618	10.33	0	0	0	13.67
3	Thursday	114519	21.67	0	2.17	0	0.17
10	Thursday	113504	6	0	4	0.50	13.33
17	Thursday	123095	16.50	0	7.50	0	0
24	Thursday	121093	24	0	0	0	0
31	Thursday	123254	17.17	0	0	0	6.83
4	Friday	126788	16.83	0	5	0	2.17
11	Friday	120387	14.33	0	5.50	0	4.17
18	Friday	98798	8.67	0	15.33	0	0
25	Friday	118876	24	0	0	0	0
7	Monday	112839	24	0	0	0	0
12	Monday	113113	23.17	0	0.83	0	0
19	Monday	114455	23.17	0	0	0	0.83
26	Monday	116943	20	0	0	0	4