IDAHO TRANSPORTATION DEPARTMENT
RESEARCH REPORT

RP 216

Improving Safety at Signalized Intersections during Inclement Weather Conditions - A Real-Time Weather-Responsive System

By
Ahmed Abdel-Rahim
Axel Krings
Ahmed Serageldin

National Institute for Advanced Transportation Technology
University of Idaho

Prepared for
Idaho Transportation Department
Research Program, Contracting Services
Division of Engineering Services
http://itd.idaho.gov/highways/research/

June 2015

DISCLAIMER

This document is disseminated under the sponsorship of the Idaho Transportation Department and the United States Department of Transportation in the interest of information exchange. The State of Idaho and the United States Government assume no liability of its contents or use thereof.

The contents of this report reflect the views of the author(s), who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official policies of the Idaho Transportation Department or the United States Department of Transportation.

The State of Idaho and the United States Government do not endorse products or manufacturers. Trademarks or manufacturers' names appear herein only because they are considered essential to the object of this document.

This report does constitute a standard, specification or regulation on report format

1. Report No. FHWA-ID-14-216	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Improving Safety at Signalized Intersections during Inclement Weather Conditions - A Real-Time Weather-Responsive System	5. Report Date June 2015
	6. Performing Organization Code KLK561
7. Author(s) Ahmed Abdel-Rahim, Axel Krings and Ahmed Serageldin	8. Performing Organization Report NIATT-N14-05
9. Performing Organization Name and Address National Institute for Advanced Transportation Technology (NIATT) P.O. Box 440901 University of Idaho Moscow, Idaho 83844-0901	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. UI-12-01
12. Sponsoring Agency Name and Address Research Program, Contracting Services, Division of Engineering Services Idaho Transportation Department 3311 West State Street Boise, ID 83707-5881	13. Type of Report and Period 05/15/12-12/31/14
	14. Sponsoring Agency Code RP 216
15. Supplementary Notes	

16. Abstract

This report presents a prototype of a secure, dependable, real-time weather-responsive system. The prototype executes 2 tasks:

1. Accesses weather information that provides near real-time atmospheric and pavement surface condition observations.
2. Adapts signal timing in response to inclement weather.

The proposed system architecture employs 2 revolutionary software design approaches:

1. Design for Survivability.
2. Software performance measurement at the task level.

Furthermore, the software design incorporates self-diagnostic techniques for fault detection and recovery to maximize the survivability and the security of the system. Minimal hardware is required for full implementation of the system as it operates and achieves its potential using current traffic signal controller and cabinet standards and technologies. As a result, it is compatible with future applications within the FHWA's connected-vehicle framework. The weather-responsive traffic signal system presented in this report serves as a major milestone in the development of secure and dependable real-time traffic control system applications.

The system was initially tested in a hardware-in-the-loop simulation environment. It was also field implemented at an intersection in the City of Moscow, Idaho. The results of the simulation environment testing indicated that the weather-responsive traffic signal system reduces the potentials for crashes while having minimal impact on delay and stops. The results showed that the system, on average, reduces the potentials for rear-end crashes, crossing crashes, and lane change crashes by 5.7 percent, 31.3 percent, and 6.3 percent, respectively. The results from the field implementation were consistent with those obtained from the simulation model. They show that, when the snow/ice signal timing plan was implemented with longer yellow interval and passage time values, the number of vehicles running the red-light during snow and ice conditions dropped significantly from 7.94 percent for passenger vehicles and 14.18 percent for trucks to 3.5 percent for passenger vehicles and 7.81 percent for trucks. This significant reduction in the number of red-light runners improves the safety of the intersection operations during inclement weather conditions.

17. Key Word: Highway Safety Improvement Program, Weather Traffic Management, Responsive Traffic Signal Systems	18. Distribution Statement Unrestricted. http://itd.idaho.gov/highways/research/archived/closed.htm and http://www.webpages.uidaho.edu/niatt

19. Security Classif. (of this report) Unrestricted	20. Security Classif. (of this page) Unrestricted	21. No. of Pages 40	22. Price

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

Technical Advisory Committee

Each research project has an advisory committee appointed jointly by the ITD Research Program Manager and ITD Project Manager. The Project Manager and Technical Advisory Committee (TAC) are responsible for overseeing projects to ensure objectives are met, reviewing deliverables, and facilitating implementation of research recommendations where appropriate . ITD's Research Program Manager appreciates the dedication of the Project Manager and TAC members in guiding this research study.

Project Manager – Brent Jennings, Idaho Transportation Department

TAC Members

Terry McAdams, (retired) - Idaho Transportation Department
Jared Hopkins - Idaho Transportation Department
Phil Rust – Ada County Highway District
Kevin Lilly – City of Moscow
Bob Koeberlein - Idaho Transportation Department
Ned Parrish -– Idaho Transportation Department

FHWA-Idaho Advisor - Lance Johnson

Acknowledgements

Special thanks are due to Dr. Michael Dixon of the University of Idaho, a co-investigator on this project, who passed away prior to the completion of this project.

Table of Contents

List of Tables

List of Figures

Executive Summary

This report presents a prototype of a secure, and reliable real-time weather-responsive system with the intent of improving the safety and efficiency of traffic signal system operations during inclement weather conditions. The prototype executes two tasks:

1. Accesses weather information that provides near real-time atmospheric and pavement surface condition observations.

2. Adapts signal timing in response to inclement weather.

Development of the prototype and the control software followed a standard systems engineering process that included six steps:

1. Review of Resources.
2. Define System Specification.
3. System Design.
4. Data Interface and Analysis.
5. Developing the Testing Environment.
6. Verification and Timing Analysis.

The prototype system architecture includes a microprocessor, external to the traffic controller that receives the weather data, analyzes the relevant data, and communicates necessary signal timing changes to the traffic controllers. Current technology supports the proposed system development, where microprocessor-traffic controller NTCIP-based communications were tested verifying that the necessary read and write capabilities are available from the microprocessor to any NTCIP-compliant traffic controller.

The weather data is accessed through the FHWA's Clarus system web interface, through ITD's weather station data server, and through a weather sensing station installed locally at the intersection. Different observation types reported in the weather data are used to determine air and surface temperature, roadway surface condition status, precipitation type and rate, and visibility level at or near the environmental sensing station. The availability and accuracy level of the weather data reported by different weather stations provided reliable estimates of the weather, road surface condition, and visibility level needed for weather-responsive traffic signal system applications.

The weather-responsive system developed in this project has five innovations.

- First, the system operates and achieves its potential using current traffic controller and controller cabinet technologies.

- Second, the system is compatible with future applications within the FHWA's connected-vehicle initiative.

- Third, minimal hardware, in addition to traffic controllers, is required for full system implementation.

- Fourth, computer driven algorithms implement traffic signal control decisions using weather data.

- Fifth, the proposed system architecture employs two revolutionary software design approaches: design for survivability and software performance measurement at the task level.

Furthermore, the software design incorporates self-diagnostic techniques for fault detection and recovery to maximize the survivability and the security of the system. Because the proposed system has very similar computational requirements to other field traffic control applications, it serves as a major milestone in the development of secure and dependable real-time traffic control systems.

The system was initially tested in a hardware-in-the-loop simulation environment. It was also field implemented at an intersection in the City of Moscow, Idaho. The results of the simulation environment testing indicated that the weather-responsive traffic signal system reduces the potentials for crashes while having minimal impact on delay and stops. The results showed that the system, on average, reduces the potentials for rear-end crashes, crossing crashes, and lane change crashes by 5.7 percent, 31.3 percent, and 6.3 percent, respectively. The results from the field implementation were consistent with those obtained from the simulation model. They show that, when the snow/ice signal timing plan was implemented with longer yellow and all red intervals and increased passage time values, the number of vehicles running the red-light during snow and ice conditions dropped significantly from 7.94 percent for passenger vehicles and 14.18 percent for commercial vehicles to 3.5 percent for passenger vehicles and 7.81 percent for trucks. This significant reduction in the number of red-light runners improves the safety of the intersection operations during inclement weather conditions. The results also show that, while vehicles were at a longer distance from the stop bar at the onset of yellow when the snow/ice signal timing plan was implemented, they were able to clear the intersection at the end of the all-red interval. These results, again confirm the potential safety effectiveness of the weather-responsive traffic signal system.

The actuated control parameters presented and tested as part of this study can be used by ITD district traffic engineers to develop snow/ice signal timing plans for isolated intersections throughout the state. The snow/ice signal timing plans can be implemented during snow/ice conditions manually or through the deployment of the weather responsive system developed and tested as part of this research. The system presented in this study is ready for implementation at the local intersection level. However, more research and development are needed before a state-wide weather responsive system can be implemented. While the state’s weather station provides a good source of weather data for the proposed weather responsive traffic signal system, using weather data from weather stations that are located at the signalized intersections improve the reliability of the weather responsive traffic signal system. These stations provide weather data that are specific to the signalized intersection and surrounding traffic network locations.

Chapter 1
Introduction

Overview

Adverse weather conditions such as rain and snow can reduce pavement friction and visibility distance, impairing the ability of drivers to operate their vehicles safely, reducing roadway capacity and significantly affecting the traffic signal system efficiency. Several empirical studies in the literature have investigated the effect of inclement weather on various signal timing traffic parameters. In terms of the effects of weather on highway safety, several studies found that weather significantly increases crash risk. A study for the Minnesota Department of Transportation reported that weather-related crash fatalities account for 17 percent of all traffic fatalities each year.(1) Another study in Montana suggests that snow increases crash risk by approximately 120, 80, 40, and 40 percent for minimal injury, minor injury, major injury, and fatal crashes, respectively.(2)

Existing studies collectively show that traffic signal timing used under normal conditions becomes problematic under adverse weather for two primary reasons. First, reductions in average speeds and saturation flow rates and the increase in start-up delays make normal signal timing unsuitable during inclement weather. Second, with reduced pavement friction and visibility, default all-red and yellow clearance intervals become unsafe as motorists are more likely to be trapped in dilemma zones. The goal of this project is to develop and implement a real-time weather-responsive traffic signal control system for the state of Idaho with the intent to improve the efficiency and safety of traffic signal operations during inclement weather.

This system will receive and use weather information from the Federal Highway Administration's (FHWA) Clarus weather data system and from Idaho's Road Weather Information Stations (RWIS) to adapt signal timing in response to inclement weather. Five innovations are necessary to fully achieve the proposed project goal.

1. First, the system will operate and achieve its potential using current traffic controller and controller cabinet technologies.

2. The system will be compatible with future applications within the FHWA's connected-vehicle initiative.

3. Minimal hardware, in addition to traffic controllers, will be required for full implementation.

4. Computer driven algorithms will implement traffic signal control decisions using Clarus data.

5. Software design will incorporate self-diagnostic techniques for fault detection and recovery to maximize security and survivability and minimize cost.

Research Objectives

The project primary objective is to develop and pilot test a real-time weather-responsive traffic signal control system for the state of Idaho with the intent to improve the efficiency and safety of traffic signal operations during inclement weather. This system receives and uses weather information from the FHWA's Clarus weather data system, the Idaho's RWIS stations, or local weather sensors to adapt signal timing in response to inclement weather.

Report Organization

This report outlines the development and testing of a real-time weather-responsive system that improves the safety and efficiency at signalized intersections during inclement weather conditions. After the introduction, system architecture and design are presented in Chapter 2. Chapter 3 presents the details of the system testing and documents the results of the simulation-based testing as well as the results of the field implementation. Finally, the study conclusions and recommendations are presented in Chapter 4.

Chapter 2
System Architecture and Design

Introduction

This chapter presents a real-time weather-responsive system. The system design incorporates state-of-the-art secure and dependable software design concepts to ensure accurate execution of two tasks. For the first task, the system accesses weather information from FHWA's Clarus data system that provides near real-time atmospheric and pavement data from participating states' environmental sensor stations (ESS). The second task adapts signal timing in response to inclement weather. Real-time control systems, especially those governing critical infrastructures such as transportation, need to be reliable and secure under normal operating conditions and survivable under abnormal conditions.

Survivability, for the purpose of traffic control applications, is defined as the capability of a control system to fulfill its mission in a timely manner, even in the presence of a component or communication failure. The proposed system employs two revolutionary software design approaches: 1) Design for Survivability, and 2) Measurement-Based Methodology (MBM). The latter is for critical applications that rely on measurements of operational systems and on dependability models to provide quantitative reliability of system performance with certain user-defined confidence levels. Furthermore, the software design incorporates self-diagnostic techniques for fault detection and recovery to maximize the survivability and the security of the system. Minimal hardware is required for full implementation of the system as it operates and achieves its potential using current traffic signal controller and cabinet standards and technologies. As a result, it is compatible with future applications within the FHWA's connected-vehicle (formally IntelliDrive) initiative. Because the proposed system has very similar requirements to other traffic control applications, it serves as a major milestone in the development of secure and dependable real-time traffic control systems.

Real-Time Weather Responsive System

Communication Architecture

The communication architecture of the proposed real-time weather-responsive traffic signal control system is shown in Figure 1. The system includes a microprocessor, external to the traffic controller that receives Clarus data, analyzes the relevant data, and communicates necessary signal timing changes to the system operator for approval. Upon approval, signal timing changes are then made in the traffic controllers. Signal timing plan adaptations include changes such as modified all-red or yellow clearance intervals or traffic signal efficiency parameters such as minimum green, maximum green, passage time as well as different coordination parameters. Suggested changes depend on multiple factors such as approach speed, pavement surface conditions, visibility, and the mode of signal operations. Current technology supports the proposed system development, where microprocessor-traffic controller communications were tested, verifying that the necessary read/write capabilities are available from the microprocessor to the controller.[3] In addition, recent advances in software design make possible fault detection and recovery for real-time in-field applications. For this prototype, the Rabbit 3000 microprocessor fulfills the role of the local processing unit shown in Figure 1.

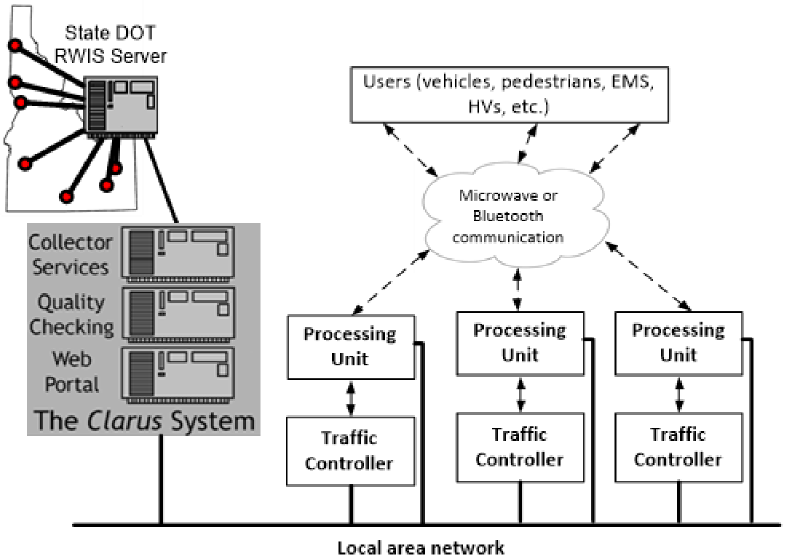


Figure 1. Communication Architecture for the Weather Responsive System

The Rabbit microprocessor is the core hardware in the system that communicates with the traffic signal controller through the Ethernet. To facilitate communications, the controller and microprocessor must follow the National Transportation Communications for ITS Protocol (NTCIP) communication standard, a family of standards for transmitting data and messages between different devices used in Intelligent Transportation Systems (ITS).[4] The Dynamic Object STMP/UDP/IP Ethernet protocol stack is used to facilitate the NTCIP-based communication between the microprocessor and the traffic controller. A computer, connected to the microprocessor through the cabinet serial connection, is used to setup and add the control logic to the microprocessor. Because the microprocessor is directly connected to the traffic signal controller through the Ethernet port, the connection is not sensitive to the cabinet configuration. However, the microprocessor requires an additional 110 volt power connection. This connection method should be possible in any NTCIP compliant controller.

The Rabbit 3000 microprocessor meets the functional requirements for real-time traffic control feedback. This type of microprocessor is designed specifically for embedded control, communications, and Ethernet connectivity. In addition, the microprocessor's processing speed is 55.5 MHz clock speed, which is more than adequate for traffic control applications. It also features a battery-operated real-time clock. In addition to the Ethernet port, it offers a 20-bit address bus, 8-bit data bus, and 3 chip select lines. Two output-enabled lines, and 2 write-enabled lines can be directly interfaced with up to 8 Flash/SRAM devices.

Inputs from the traffic signal and Clarus weather sensors, signal status, phase timing plan, next phase, and phase omit data are required for the Rabbit to adapt timing plans and disseminate the feedback. Table 1 shows the data accessibility of different relevant objects within an NTCIP compliant traffic controller.

It should be noted that all NTCIP dynamic objects related to phase status are read-only. As a result, termination of a phase is not a feasible control feedback option. Instead, phase operations must be influenced by changing parameter values such as minimum green, maximum green, passage time, etc. A program, installed on the Rabbit, governs its operations and is programmed using the Dynamic C® software development system. Dynamic C is an integrated C compiler, editor, loader, and debugger fashioned for the Rabbit microprocessor. There are two basic sections in the code. The first section is developed for communications, sending data requests, receiving data, and sending control feedback to the operator and to the controller. The second section is written to process data, determine control decisions, and send control feedback.

Software Design

The proposed software architecture employs two revolutionary new approaches:

1. Design for Survivability.
2. Measurement-Based Methodology (MBM) for Embedded Systems.

Whereas the concepts have been discussed in the fault-tolerance and security community for almost a decade, implementations are limited to academic prototypes, none of which were in traffic signal systems. The main reason is that most systems that would benefit from these approaches already exist and it is uneconomical to retrofit these two principles, i.e., their very principle is based on integration and not retrofitting.

The project described here could serve as a major milestone in the development of safe and secure transportation systems. First, it is sufficiently small in scope to utilize both approaches in a manageable way. Second, the application is part of a critical infrastructure, therefore justifying the additional complexity and effort. Most importantly, most applications that have considered high levels of fault-tolerance have been in the area of ultra-reliable systems, which typically include systems like primary flight control, or military applications. However, even those applications are only now realizing the need for survivability in addition to fault-tolerance.

The architecture of the proposed work is a fundamental building block in a highly networked and interactive communications system. As such it will be exposed to all faults that may occur locally or via the network, ranging from benign component failures to malicious cyber threats. Due to the fact that this is a safety critical system, the design process associated with ultra-reliable real-time systems design must be used. As a result, we propose using the design philosophy called "Design for Survivability" to incorporate fault tolerance in a more general way as it not only considers component or software faults, but also faults associated with malicious act, i.e., maliciously induced faults.(5,6) In this way, the project is based on a measurement-based methodology for survivability of transportation control system components.

"Design for Survivability" is an approach that has much in common with "Design for Testability". As integrated circuits became larger, exhaustive testing became infeasible, i.e., the number of test scenarios needed to test circuits became intractable. As a result, it was realized that circuits had to be designed for testability. As systems became increasingly complex and difficult to analyze, the notion of designing for survivability, i.e., integrating the mechanisms that aid survivability into the system (rather than as an add-on feature), became a natural extension, analogous to design for testability.(5) As a result, to achieve this level of survivability, the proposed system software design employed testing in the form of system measurements and self-diagnostics.

Our proposed system needed the basic capability of using and generating data, i.e., data imported from Clarus as well as the potential to serve as a sensor (i.e., data provider) for Clarus. To accommodate this need, the system required a secure interface, capable of dealing with basic fault types. There are too many fault sources to list them individually and exhaustively. Therefore the notion of fault models is used, capturing the behavior of a fault, i.e., a fault can produce an error that then can lead to a failure. The diversity of faults and their consequences on a system have been the primary motivator for the definition of fault models. A fault model addresses the behavior of the faults and specifies the redundancy levels required to tolerate a single fault type or a mix of fault types. Many different fault models have been proposed over the years ranging from the simple models that make no assumptions about the fault behavior, to hybrid fault models considering multiple fault behavior. The latter considers a mix of faults ranging from benign, symmetric to asymmetric faults, with potential transmissive and omissive behaviors.(7,8,9)

These latter 5-fault models constitute the basis for the faults addressed in the proposed system and that are considered in the communications with Clarus. Omission faults will be emphasized, because communication with Clarus may be interrupted. Furthermore, value faults (symmetric and asymmetric) such as infeasible or incorrect input or output data were also deemed important, since any of such faults have the potential to decrease safety. Benign and transmissive faults were included too, were also applicable but were the least critical.

Rather than specifying each of the functionalities of the software, we want to focus on the software architecture as it addresses design for survivability and MBM. There are several key technologies incorporated in these two approaches, including: functional software specification, measurement-based certification of normal and abnormal operation, adaptability, diagnosability, real-time predictability and fail-safe behavior. In short: all the ingredients to run, observe, analyze, and reconfigure a system.

The most important aspect of the software architecture is the derivation/adaptation of the MBM introduced in and refined in to ensure properties of reliability, security and survivability.(10,11) Intuitively, the application is defined as a basic set of operations, which are expressed by a collection of functionalities. These functionalities are implemented with software modules, e.g., C functions, and instrumented (via instrumentation telemetry) in a way that allows measuring the behavior of the operations, functionalities and modules in real-time. This becomes extremely useful when studying the behavior of individual functionalities during execution. As has been demonstrated in previous work with the behavior of networked systems under attack, normal executions of functionalities can be captured like a finger print of that functionality (called functional profiles).(10)

Any deviation from such a profile can then be interpreted as an unusual execution. This in turn allows for responsive measures (e.g., changing the execution state, re-executing a functionality or system reconfiguration) as defined by a contingency management system. The basic operation of this approach is shown in Figure 2.

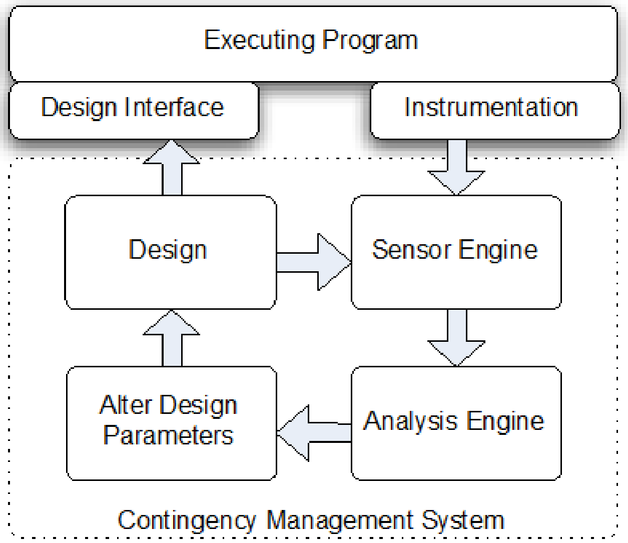


Figure 2. Weather Responsive System Operation Flow

The executing program is observed via the instrumentation telemetry. The feedback-loop of:

1. Observing.
2. Analyzing.
3. Changing Parameters.
4. Controlling the Software Design or Operation is critical during software design and later during its operation.

In the latter case, it allows implementation of the survivability measures upon detection of deviations from certified operation, e.g., unusual or undesired operation.

The overall system architecture is comprised of multiple components, the executing program and the contingency management system. The sole purpose of the latter is to watch the execution in real-time and react to unwanted changes, as they would occur as the result of system components malfunctioning or unwanted manipulations of the system by intruders and/or hackers.

Should multiple systems be deployed in proximity, then the principle of spatial redundancy can be used to tolerate failures and malicious attempts to manipulate the system. For this situation, agreement algorithms can be used to eliminate the impact of incorrect values and data. For instance, if the system needs to agree on values that represent Clarus data, then exact agreement algorithms can be used, e.g., the early stopping agreement as documented in Krings.(12) If there are real-valued control parameters that have to be agreed upon, approximate agreements can be used.(9,13)

Software Architecture

Figure 3 shows an overview of the software architecture and its interface to Clarus. Shaded blocks indicate the hardware interfaces. The Network Interface represents the connection to the Internet. Because the signal control system has its own data representation, the Clarus data has to be translated in the Clarus Data Conversion Interface. However, the kind of Clarus data required depends on the enabled control algorithm. These algorithms are modular units in the Algorithm Engine. Different algorithms may have different data needs, as specified in the Data Type Definitions. Thus the Clarus Data Management Engine is the logical interface between the algorithm's data requirement and the Clarus database. The Traffic Controller Adaptation Engine becomes the executive of the control algorithm.

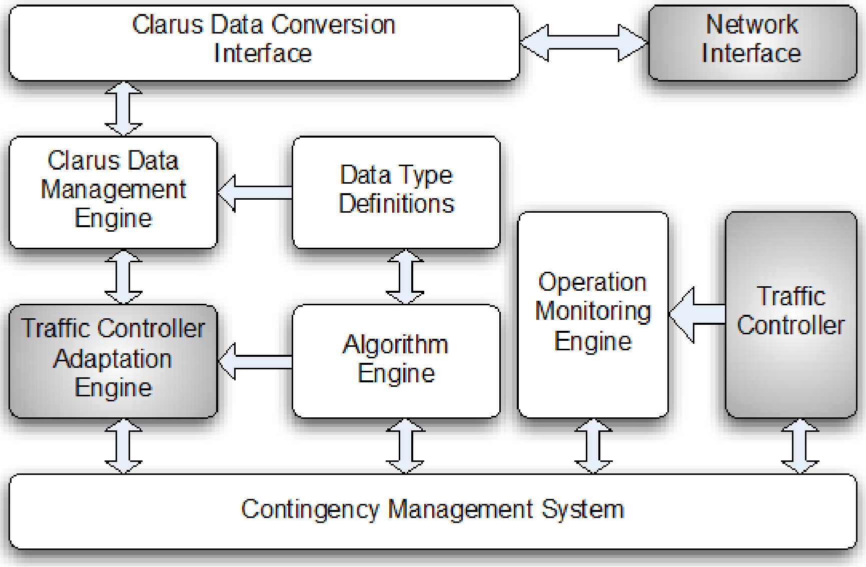

Clarus Data Conversion Interface
Network Interface
Clarus Data Management Engine
Data Type Definitions
Operation Monitoring Engine
Traffic Controller
Traffic Controller Adaptation Engine
Algorithm Engine
Contingency Management System

Figure 3. Software Architecture

Survivability measures during the design and operation of the system are centered on the Traffic Controller and the Operation Monitoring Engine, which interfaces to the controller via the instrumentation telemetry, i.e., the monitoring engine receives its sensor data from the executing controller. The adaptability and recovery from any unintended or maliciously induced operations/profiles is determined by the survivability policy and is handled by the Contingency Management System. Whereas the block diagram of Figure 3 suggests a high level of complexity, the goal of the project is to operate in a low-complexity environment. The relatively small size of the system makes it a perfect candidate to apply the survivability and the measurement-based methodology effectively.

Survivability Architecture

General Principle and Definitions

The notation and general execution model described below is adapted from to suit the more deterministic execution environment of this application.(11) The Rabbit executes a set of operations O with cardinality $|O|$. These operations constitute the operational machine that defines the operations of the application without implementation considerations. The transition from one operation to another marks an *operational epoch*. Each operation o_i uses one or more functionalities f_j from a set F of functionalities with cardinality $|F|$. Similar to the operational epoch, the *functional epoch* is defined by transitions from one functionality to another. Functionalities are implemented using program modules, e.g., Dynamic C code modules. The set of modules M of cardinality $|M|$ is thus the implementation of the functionalities in code. Thus the software that is written to implement the system is nothing but a collection of program modules derived from the functional specifications so that the system performs the operations defined by the system functionalities.

The relationship between operations, functions, and modules is defined by a graph G^{OFM}, where the superscript indicates that the graph maps from O to F and M. The example depicted in Figure 4 shows three operations o_1, o_2 and o_3. The operations utilize specific functionalities, e.g., o_1 uses functionalities f_1 and f_2. Incidentally, f_2 is also used by o_3. The functionalities are implemented by Dynamic C modules, e.g., f_3 is implemented by module m_4, whereas f_4 is realized by m_4, m_5, and m_6.

System Operation Profiles

During the operation of the system, and with proper instrumentation of the software, one can get a "life" picture of how the system is performing in real time, e.g. what the execution of a typical operation looks like, how often functionalities are called by a specific operation, what mix of functionalities is instantiated over a certain window of observation, or how often certain modules get called during a time interval. All of this information is captured in profiles. The frequency spectrum of operations, functionalities and modules define the operational, functional, and module profile respectively. These profiles will be used later to define certified operations. Leaning on the notation of we will use letters u, q and p for operational, functional and module profiles, respectively.(11)

Let u_l denote the probability that the system is executing operation o_l. Then $u =[u_1, u_2, …., u_{|O|}]$ is the operational profile of the system. During execution of the system, we are interested in observing the operational profile over n epochs. This observed profile is $\hat{u} =[\hat{u}_1, \hat{u}_2, …., \hat{u}_{|O|}]$ where $\hat{u}_i= c_i/n$ is the fraction of system activity due to operation o_i and c_i is the count, i.e., frequency or spectrum of invocation, of o_i. As the system activity is continuously monitored, which implies that operational profiles are generated and analyzed, we want to keep track of these profiles. Let $\hat{u}^k$ denote the k^{th} operational profile. Thus $\hat{u}^k$ is the k^{th} observed operational profile, observed over n operational epochs, which was preceded by $\hat{u}^{k-1}$, observed over the previous n operational epochs, and so forth.

If we consider *m* sequences of n epochs each, we can define a centroid $\bar{u} = [\ \bar{u}_1, \bar{u}_2, ..., \bar{u}_{|O|}]$, where:

$$\overline{u}_i = \frac{1}{m} \sum_{j=1}^{m} \hat{u}_i^j$$

Figure 4. Equation: Centroid *ū*

and the distance d_k from $\hat{u}^{\ k}$ to centroid $\bar{u}$ is defined by:

$$d_k = \sum_{i=1}^{n} (\overline{u}_i - \hat{u}_i^k)^2$$

Figure 5. Equation: Distance d_k from $\hat{u}^{\ k}$ to Centroid $\bar{u}$

Whereas the example in Figure 6 shows the relationship between operations, functionalities, and modules, it does not contain any information about dependencies of operation in *O*, functionalities in *F*, or modules in *M*.

The relationship between operation is defined by graph $G^O = (O,\angle)$, where $\angle$defines a partial order relation on the operations in *O*, i.e., if o_j depends on o_i then $(o_i,o_j) \in \angle$. In the example of Figure 4, if o_1 is the operation "obtain data", o_2 is "analyze data", and o_3 is "adjust controller", then the logical dependencies among the operations are $o_1 \angle o_2$ and $o_2 \angle o_3$. Any violation of the partial order indicates a problem, i.e., an error, in the control flow of the program.

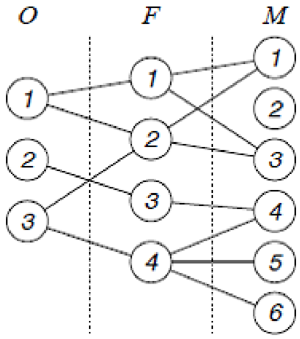


Figure 6. Operation-Function-Module Mapping (O x F x M)

We define similar graphs for functionalities and modules, however, the precedence relation in those cases is a general precedence relation and not necessarily a partial order, e.g., the graph may not be acyclic. Thus $G^F = (F,<)$ and $G^M = (M,<)$ are the graphs defining calling relationships between functionalities and modules respectively. It should be noted that G^M is the static call graph of modules in *M*, i.e., the graph that indicates what module in the program can call what module. Furthermore, the difference in precedence relations should be noted, i.e., $\angle$denotes a partial order relation, whereas < in general does not. The operational, functional, and module dependency graphs are used to detect invalid transitions. These are transitions that derive from unperceived executions as the result of programming errors, input data errors, or code manipulations.

Dispatching Model

The Rabbit system uses a single processor in which multitasking is implemented using a model defined by costatements, which are different from conventional multi-processing or multi-threading models. If one wants to compare it to the aforementioned models, then perhaps user-level threads probably have the most resemblance. Simplistically speaking, the system executes one costatement at a time. Each costatement has a statement counter, i.e., a program counter, which indicates which instruction of the costatement will executed when it gets a chance to run. Execution is switched from one costatement to the next when the currently executing costatement yields. Therefore, the model is based on good behavior. The state of a costatement is called a costate. In the discussions to follow, the terms costatement and costate will be used interchangably.

A model with such task-switching properties executes deterministic, i.e., a task switch is explicitly demanded by the currently executing tasks: the costatement. On the other hand this means however that it is possible for a costatement to cause starvation by not yielding. To resolve such situations, mechanisms like watchdogs and timer interrupts can be used.

Operations, functionalities, and modules are implemented by (or called from within) costatements. It is possible to exactly determine the functionality and module that is being executed on behalf of a specific operation, as will be described in the discussion on the current state of the system below.

Run-time Monitoring

Instrumentation

There are three types of instrumentation: for operations, functionalities and modules. For each the specific steps are described below. However, it should be noted that one can have a mix of instrumentations. For example, if a module also indicates the start of a functionality, then this instrumentation has to be included as well. Thus, in the most complicated case we could have to instrument the beginning of an operation, then the beginning of a functionality, and then a module. Furthermore, the instrumentation has to be in that order.

Operation Instrumentation:

When entering an operation o_i in costate α =*ActiveCostateID* the following tasks are performed:

1. Check for violation of partial order relation in G^O.

2. Update $S[\alpha]$ to indicate o_i is now the current operation, i.e., $S[\alpha]$=[*o_i*,-,-], where - indicates no change.

3. Increment the frequency count $c^o_i[\alpha]$$ to account for the instantiation of o_i.

Functionality Instrumentation:

When entering a functionality f_i in costate α = *ActiveCostateID* the following tasks are performed:

1. Check for violation of partial order relation in G^F.

2. Check for violation of mappings in G^{OFM}, i.e., determine if the execution of f_i is consistent with the operations in the graph.

3. Update $S[\alpha]$ to indicate f_i is now the current functionality, i.e., $S[\alpha]$ = [-,f_i,-].

4. Increment the frequency count $c^f_i[\alpha]$ to account for the instantiation of f_i.

Module Instrumentation:

When entering a module m_i in costate α = *ActiveCostateID* the following tasks are performed:

1. Check for violation of precedence relation in G^M.

2. Check for violation of mappings in G^{OFM}, i.e., determine if the execution of m_i is consistent with the operation and functionality in the graph.

3. Update $S[\alpha]$ to indicate m_i is now the current module, i.e., $S[\alpha]$ = [-,-,m_i].

4. Increment the frequency count $c^m_i[\alpha]$ to account for the instantiation of m_i.

Experimental Results

A prototype has been built based on a Rabbit MiniCore RCM5700, which incorporates the Rabbit 5000 microprocessor with integrated 10/100Base-T Ethernet functionality and 128KB of on chip SRAM. The Rabbit is running Dynamic C version 10.5.4, which has been instrumented to allow operation, function and module profiling. Furthermore, at each level of abstraction the precedence constraints can be validated. The current system utilizes 395 modules, of which 177 are written in Dynamic C and 218 in assembler code. All Dynamic C modules were instrumented. A partial sample profile of the system is given in Figure 7 in which 46 significant modules are represented in 4 costates.

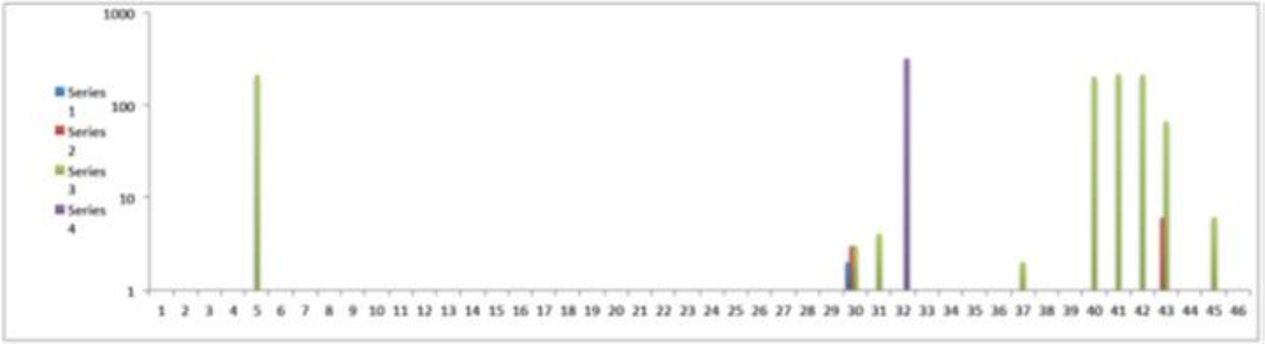


Figure 7. Sample Frequency Counts

Current efforts focus on generating profiles and evaluating them in real-time in order to determine reasonable certification thresholds. Whereas instrumentation for operation and functionalities was implemented, actual testing and evaluating has focused mainly on modules. This strategy was used in order to optimize the learning curve, given the realities of the project duration. Furthermore, some features of dependency modeling have turned out to be more challenging than originally foreseen. Specifically, instrumenting Dynamic C library modules has been limiting, in that assembler modules could not be instrumented. This however had implications when such modules call Dynamic C modules, i.e., call graph dependencies when assembler modules call Dynamic C modules cannot be validated. An approach was taken to deal with this problem by treating validation of violations as a sensor input, similar to the frequency counters. Thus, dependency violations thresholds have to be evaluated.

Software Design Conclusions

Real time monitoring the executions of the operational and functional machines as well as modular profiling have been explored in order to aid in:

1. The design of embedded systems.
2. In the reconfiguration upon detecting deviation from certified behavior.

The formal model was introduced and expanded to take advantage of the decrease in non-determinism of executions in the costate task management paradigm.

Chapter 3
Field Implementation and Testing

Introduction

Adverse weather conditions have a significant effect on drivers and driving conditions. Research has suggested that parameters like saturation flow rate, average speed, and acceleration and deceleration rates, among others, vary in heavy rain, snow, or ice conditions when compared to dry weather conditions. This chapter provides a comprehensive evaluation of the potential safety and operational benefits of weather responsive traffic signal systems. Microscopic simulation modeling and field implementation and testing were used to assess the potential safety and operational benefits of adjusting different signal timing parameters during inclement weather conditions. In the simulation modeling, traffic safety benefits were assessed through the use of surrogate measures such as the number and type of conflicts due to weather effects. The weather responsive traffic signal system developed as part of this project was implemented at a signalized intersection in Moscow, Idaho. Data was collected before-and-after the field implementation of snow and ice signal timing plans to assess the potential safety benefits of the weather responsive traffic signal system.

Simulation Model Testing

The effectiveness of the weather-responsive traffic signal system was tested first on a hardware-in-the-loop microscopic traffic simulation environment. An Econolite ASC/3-2100 controller was used as part of the hardware-in-the-loop simulation. The controller was connected to the VISSIM microscopic simulation model through a controller interface device (CID) (Figure 8). The controller provided real-time actuated control to an isolated intersection in the VISSIM simulation model. The car following parameters in the VISSIM microscopic simulation model we adjusted to represent traffic operations during snowy and icy conditions. The safety impact of different traffic control plans was assessed using the FHWA's Surrogate Safety Assessment Model (SSAM).[14] Full details of the model development, calibration, and validation can be found in Ricks.[15]

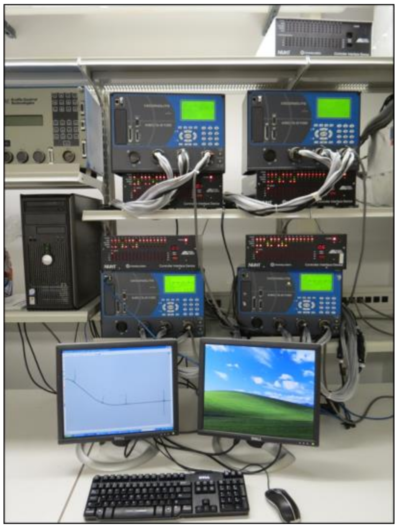


Figure 8. Hardware-in-the-Loop Microscopic Simulation Model

For normal (clear) weather conditions, the minimum green time, the yellow interval, the all red interval, and the vehicle extension time values were set to 5 seconds, 3 seconds, 1 second, and 2 seconds, respectively. During snow weather conditions, these values were adjusted to 6.5 seconds, 4.0 seconds, 1.3 seconds, and 3.2 seconds, respectively. For ice conditions, these values were adjusted to 8 seconds, 4.5 seconds, 1.5 seconds, and 3.6 seconds, respectively. Four cases were considered in the analysis, snow and ice conditions with no timing change (NTC) and snow and ice conditions with signal timing parameters changed (TC). Each case was run 10 multiple runs with different random seed numbers to generate different traffic conditions and to account for the stochastic nature of the traffic. Three volume levels were considered in the analysis: high traffic flow level (1,000 vehicles per lane per hour), moderate traffic flow level (500 vehicles per lane per hour), and low traffic flow level (300 vehicles per lane per hour). The minor road traffic volume level was kept constant at 400 vehicles per hour per lane.

The isolated intersection experienced little change in delay as a result of snow and ice signal timing plans implementation. Figure 9 and Figure 10 show the average delay per vehicle for each volume scenario for snow, and ice conditions respectively. Each figure has a side-by-side comparison of the delay with no timing change (NTC) and with the timing change (TC). For high and moderate traffic flow levels, the average delay per vehicle increased by an average of 5.1 percent when the snow and ice traffic control plans were implemented. This increase can be attributed to the increase in the duration of the yellow, all red, and vehicle extension times. The results clearly suggest that implementing snow and ice signal timing plans has marginal effect on average delay at the intersection.

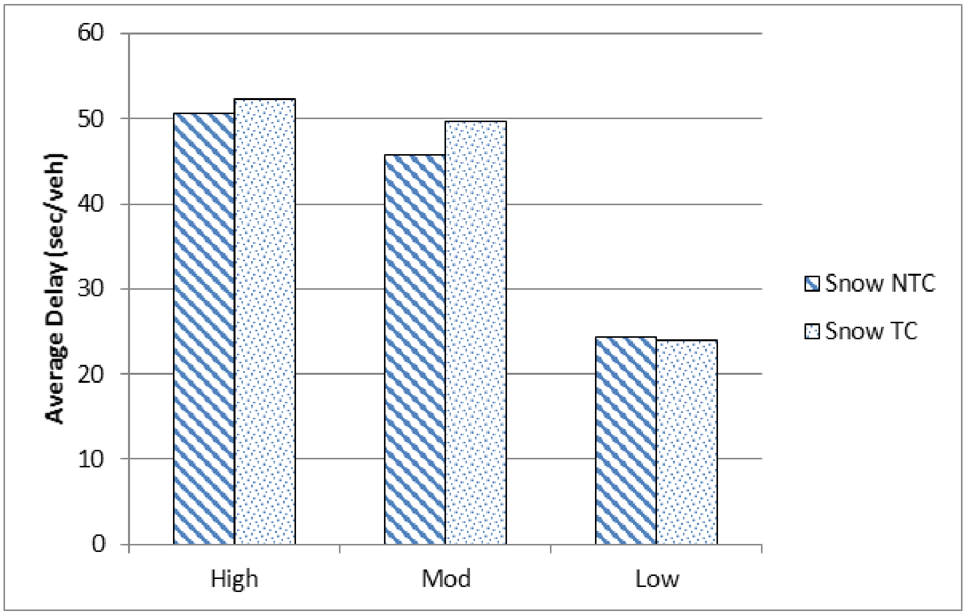


Figure 9. Isolated Intersection Delay Under Snowy Conditions

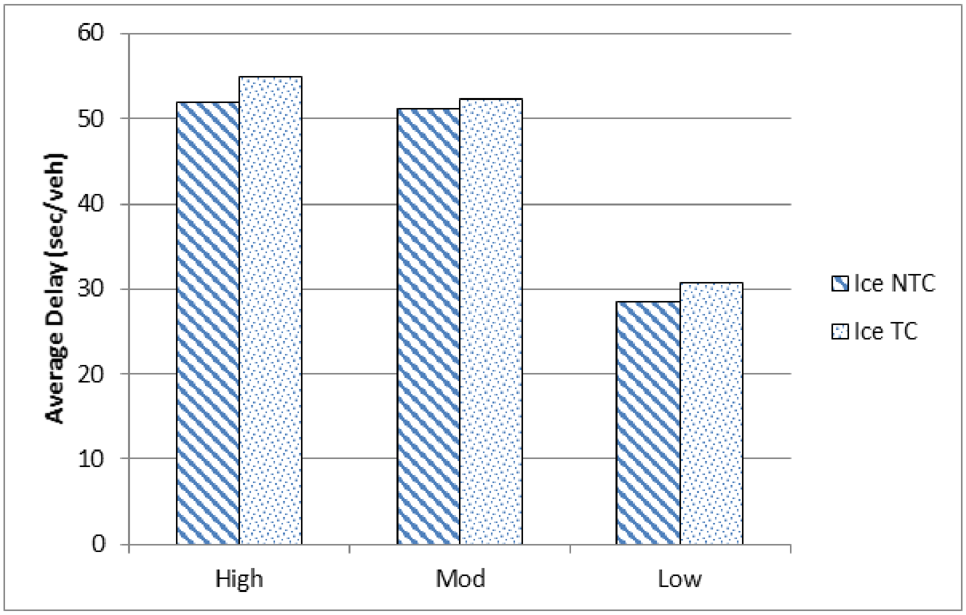


Figure 10. Isolated Intersection Delay Under Icy Conditions

The results of the average number of stops at the intersection (Figure 11 and Figure 12) showed a trend similar to that of the average delay. During snowy conditions, the average number of stops at the intersection decreased by an average of 1.1 percent. For ice conditions, the average number of stops during high and moderate traffic flow levels increased by an average of 3.4 percent. Again, the results suggest that implementing snow and ice signal timing plans has marginal effect on the average number of stops at the intersection.

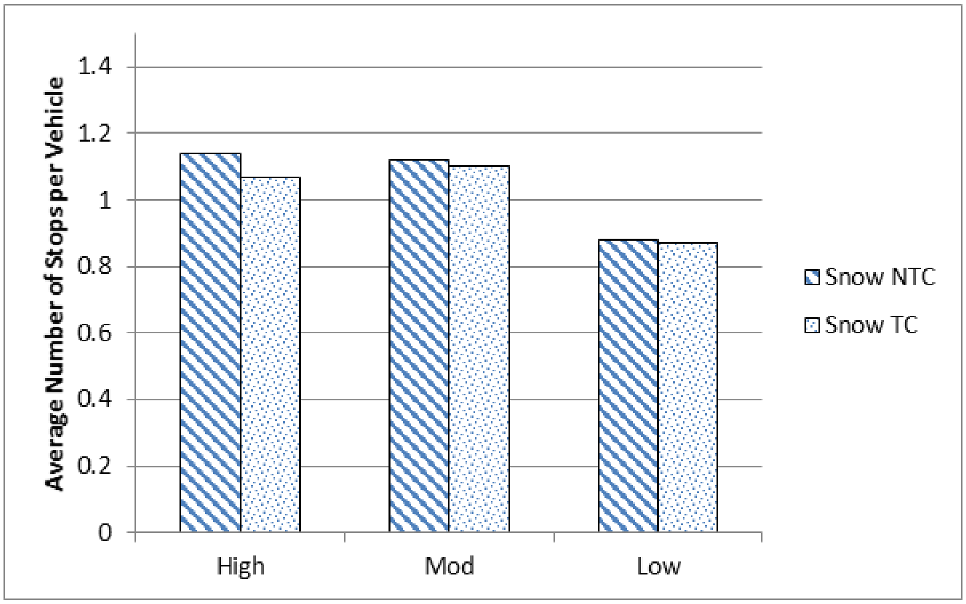


Figure 11. Isolated Intersection Number of Stops Under Snowy Conditions

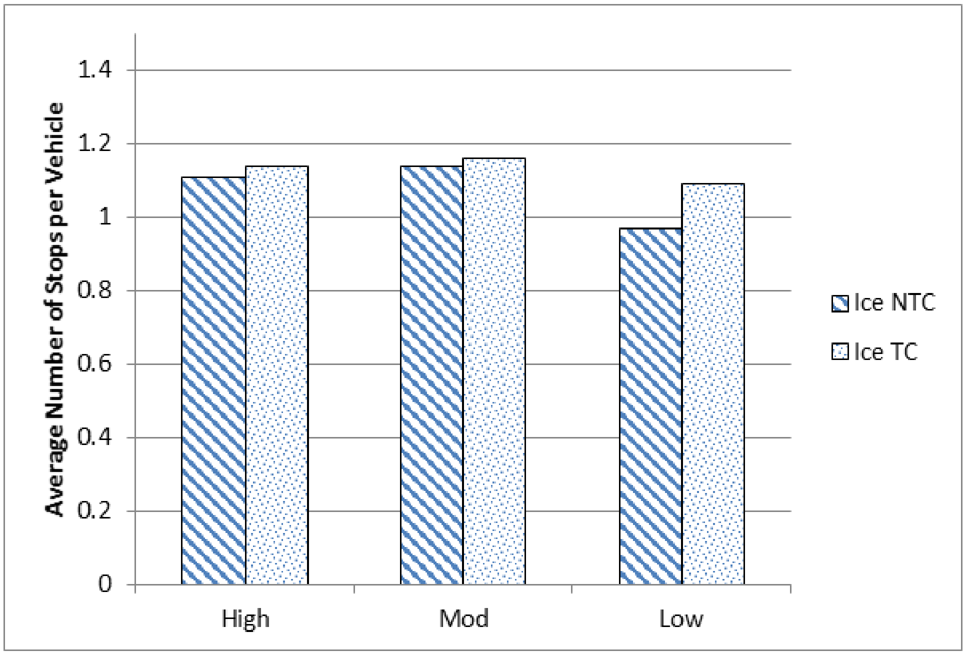


Figure 12. Isolated Intersection Number of Stops Under Icy Conditions

Figure 11 and Figure 12 show the crash rates for snowy and icy conditions, respectively, as reported by the SSAM tool. The SSAM tool compares the time distance between vehicles in the VISSIM output files and compare these distances to a pre-set time-to-collision (TTC) value. In this analysis, TTC was set to 0.1 seconds which is the resolution of the VISSIM simulation model output. If the time distance between 2 vehicles is less than 0.1 seconds, the SSAM tool will report this as a possible conflict of crash. It should be noted here that the crash rates reported by the SSAM tool are not a prediction of how many actual accidents can be expected at the intersection since SSAM cannot accurately depict a real crash rate from VISSIM data. Rather, they provide an assessment of the relative safety of different scnarios. The SSAM crash rate results, presented in Figures 13 and 14, show that implementing snow and ice signal timing plans has led to an overall decrease in all accident types. The results are summarized in Table 1. On average, there was a decrease in rear-end crashes, crossing crashes, and lane change crashes by 5.7 percent, 31.3, percent, and 6.3 percent, respectively.

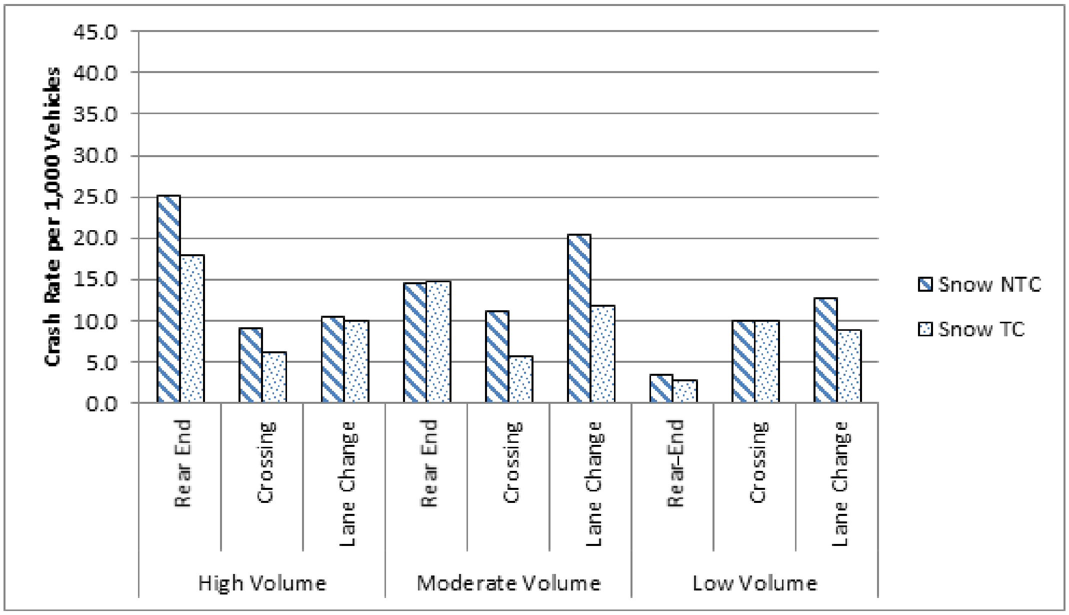


Figure 13. Isolated Intersection Crash Rate Under Snowy Conditions

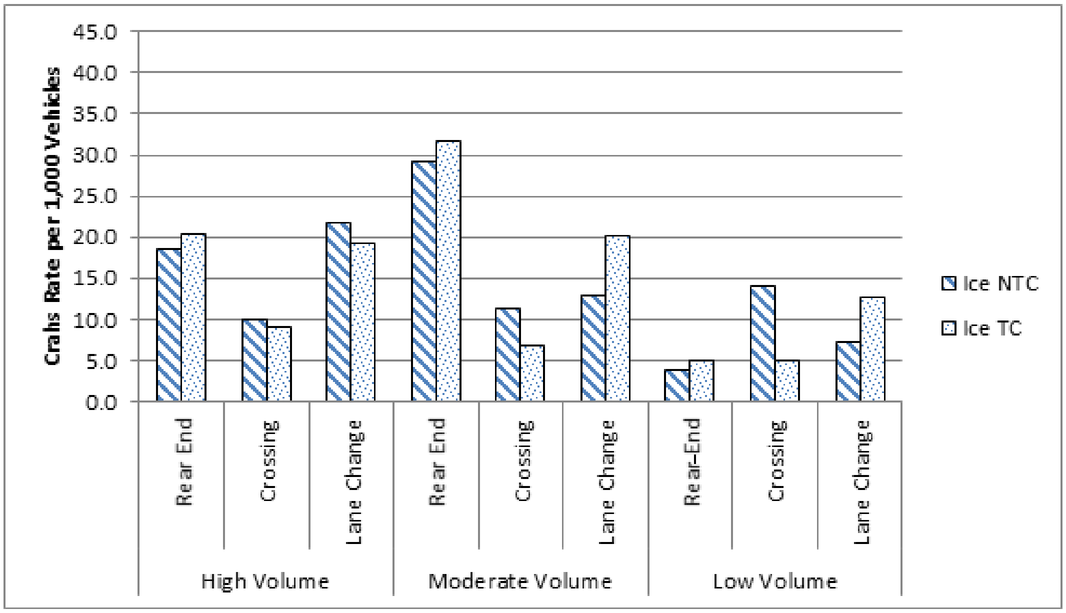


Figure 14. Isolated Intersection Crash Rate under Icy Conditions

Table 1. Percent Reduction in Total Crash Rate

	Rear-End	Crossing	Lane Change
NTC Crash Rate (per 1,000 vehicles)	135	111	99
TC Crash Rate (per 1,000 vehicles)	128	77	92
Percent Change	5.7%	31.3%	6.3%

Field Implementation

The real-time weather responsive traffic signal system was implemented at the US 95 and Palouse Rive Drive Intersection in Moscow, Idaho (Figure 15). The intersection has a Closed Circuit TV (CCTV) camera that enabled field data collection before-and-after the implementation of the system. Three different implementation architectures were designed and tested as part of this project. Figure 16 shows the implementation architecture using weather data from the FHWA's Clarus server. Figure 17 shows a similar architecture where the weather data is obtained through ITD RWIS's server rather than the Clarus server. In both these architectures, the weather data is obtained through a connection over the internet. To address ITD staff security concerns, a two-layer hardware-based firewall along with a control switch was developed to ensure full protection of the ITD network. Full details of the firewall design and implementation are provided in a separate report to ITD staff. This secure firewall design is not specific to the responsive weather system application. It can be used to secure ITD network in applications where the network needs to be connected with other external networks.

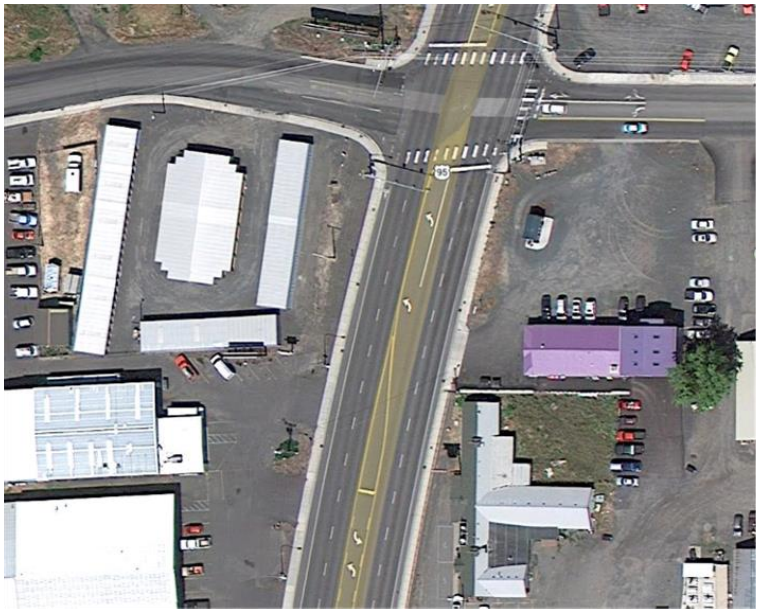


Figure 15. US 95 and Palouse River Drive Intersection in Moscow, Idaho

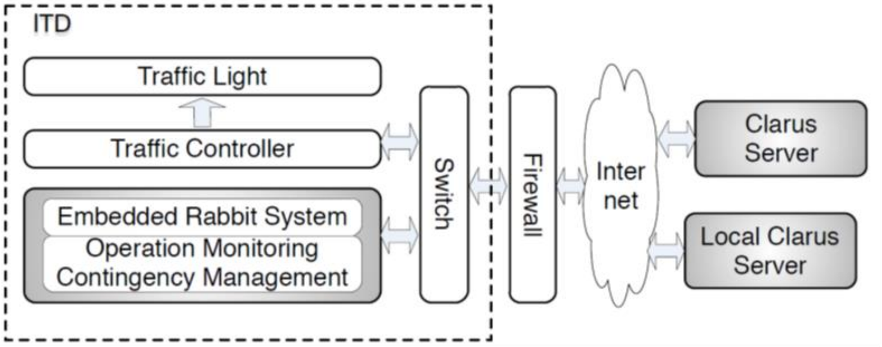


Figure 16. Implementation Architecture Using Weather Data from the FHWA Clarus Server

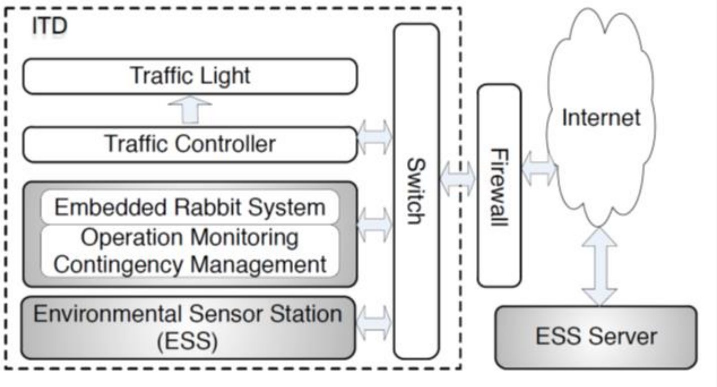


Figure 17. Implementation Architecture Using Weather Data from ITD RWIS

The third implementation architecture presented in Figure 18 was the architecture implemented in the field. In this architecture, weather data are obtained directly from a local weather station installed at the intersection. In this architecture, all the data are exchanged within ITD network eliminating any network security concerns. The other advantage of this system is that the weather data are obtained locally at the intersection which improves the reliability of the weather data used in the weather-responsive system. Figure 19 shows the details of the installation of the local weather station at the US 95 and Palouse River Drive Intersection.

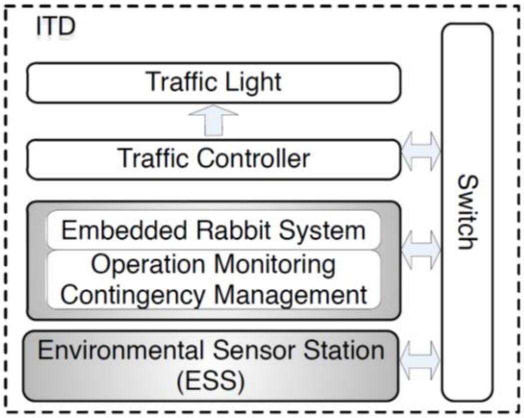
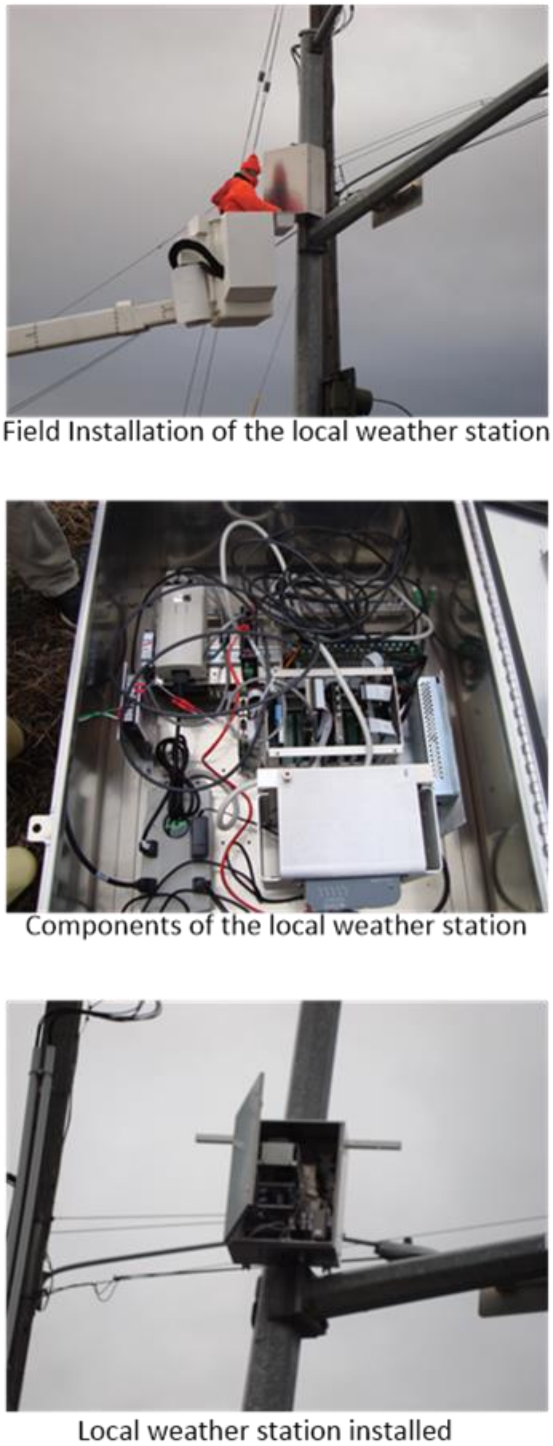


Figure 18. Implementation Architecture Using Weather Data from Local Weather Station



Field Installation of the local weather station

Components of the local weather station

Local weather station installed

Figure 19. Installation of Local Weather Station at the US 95 and Palouse River Drive Intersection

Field Implementation – Data Collection and Analysis

Data collection took place during the spring of 2013 and 2014. Live feed from the CCTV camera was used to collect data focusing on vehicles travelling northbound on US 95 at the intersection. Cones and other permanent markers were placed at the intersection approach every 20 feet to facilitate distance measurements. The signal heads that serve the approach were visible in the CCVT view and allowed for determination of the start and end of the yellow phase interval for the approach. For each vehicle, two distances were measured from the CCTV views (Figure 20):

X_y = the distance from the vehicle's front point to the stop bar at the onset of the yellow interval,

X_r= the distance from the vehicle's rear point to the stop bar at the end of the all-red interval.

In addition to these two distances, data was collected on whether the vehicle ran the red-light or not. Data was collected during snow and ice conditions for two cases: with signal timing adjustments and without signal timing adjustment.

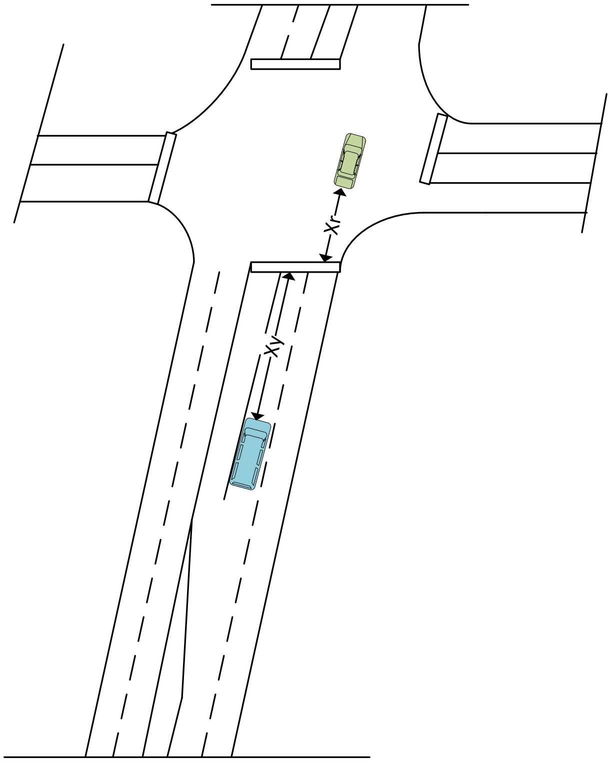


Figure 20. Field Data Collection at the Intersection Approach

While the weather responsive system has the capability of changing signal timing parameters at the controller directly through the NTCIP commands, this capability was not utilized in the field implementation. Instead, the system was modified to send a text message to ITD staff when it confirms snow or ice conditions. After receiving the message, ITD staff manually implemented the snow/ice signal timing plan at the intersection. Similarly, once the system confirms that the snow/ice conditions are cleared, the weather responsive system would send a text message to ITD staff, who would deactivate the snow/ice signal timing plan and implement the normal signal timing plan. The approach was chosen to ensure full control of ITD staff on system implementation during testing.

Data was collected for a total of 2,368 vehicles that traveled through the intersection approach during snow/ice conditions in spring 2013 and spring 2014. Of these vehicles, 521 travelled through the intersection approach when the snow/ice signal timing plan was implemented. A summary of the results is presented in Table 2.

Table 2. Summary of Field Data Collection Results

Snow/Ice Signal Timing Plan	Vehicle Type	Number of Vehicles	Percent of Vehicles Who Run the Red-Light	Average X_y (feet)	Average X_r (feet)
Not Implemented	Passenger Vehicle	1,713	7.94	189.4	60.4
	Trucks	134	14.18	183.4	58.1
Implemented	Passenger Vehicle	457	3.50	210.8	62.3
	Trucks	64	7.81	201.3	61.8

The results shown in Table 2 clearly demonstrate the safety effectiveness of the weather-responsive traffic signal system developed and tested as part of this project. When the snow/ice signal timing plan was implemented, the number of vehicles running the red-light during snow and ice conditions dropped significantly from 7.94 percent for passenger vehicles and 14.18 percent for trucks to 3.5 percent for passenger vehicles and 7.81 percent for trucks. This significant reduction in the number of red-light runners improves the safety of the intersection operations during inclement weather conditions. The results also show that, while vehicles were at a longer distance from the stop bar at the onset of yellow when the snow/ice signal timing plan was implemented, they were able to clear the intersection at the end of the all-red interval. These results, again confirms the safety effectiveness of the real-time weather-responsive traffic signal system.

Chapter 4
Conclusion and Recommendations

A reliable and secure weather responsive traffic control system for a signalized intersection was developed and tested as part of this project. The system utilizes data from weather stations or sensors to determine the status of the pavement surface condition and update signal timing plan parameters accordingly. Minimal hardware requirements exist for the prototype, where the off-the-shelf microprocessor, access to power, and a connecting cable are entirely sufficient. The prototype design is such that it would function for any field traffic control application, where the overall process can be distilled to predictable tasks. Current traffic control technology supports the proposed system development, where microprocessor-traffic controller NTCIP-based communications were tested verifying that the necessary read and write capabilities are available from the microprocessor to any NTCIP-compliant traffic controller.

Development of the prototype followed a standard systems engineering process that included six steps: review of resources, define system specification, system design, data interface and analysis, developing the testing environment, and verification and timing analysis. The weather data is accessed through a subscription to the FHWA Clarus system web interface, ITD RWIS data server, and local weather sensors installed locally at the intersection. The overall system architecture is comprised of multiple components, the executing program and the contingency management system. The sole purpose of the latter is to watch the execution in real-time and react to unwanted changes, as they would occur as the result of system components malfunctioning or communication failure. Survivability measured during the design and operation of the system are centered on the Operation Monitoring and Contingency Management System, which interfaces with the software system via the instrumentation telemetry. The adaptability and recovery from any unintended or maliciously induced operations/profiles is determined by the survivability policy and is handled by the Contingency Management System. Because the proposed system has very similar computational requirements to other field traffic control applications, it serves as a major milestone in the development of secure and dependable real-time traffic control systems.

The system was initially tested in a hardware-in-the-loop simulation environment. It was also field implemented at an intersection in the City of Moscow, Idaho. The results of the simulation environment testing indicated that the weather-responsive traffic signal system reduces the potential for crashes while having minimal impact on delay and stops. The results showed that the system, on average, reduces the potentials for rear-end crashes, crossing crashes, and lane change crashes by 5.7 percent, 31.3 percent, and 6.3 percent, respectively.

The results from the field implementation were consistent with those obtained from the simulation model. They show that, when the snow/ice signal timing plan was implemented, the number of vehicles running the red-light during snow and ice conditions dropped significantly from 7.94 percent for passenger vehicles and 14.18 percent for trucks to 3.5 percent for passenger vehicles and 7.81 percent for trucks. This significant reduction in the number of red-light runners improves the safety of the intersection operations during inclement weather conditions. The results also show that, while vehicles were at a longer distance from the stop bar at the onset of yellow when the snow/ice signal timing plan was implemented, they were able to clear the intersection at the end of the all-red interval. These results, again confirm the potential safety effectiveness of the weather-responsive traffic signal system.

Based on the results of this study, the following recommendations can be made:

- The actuated control parameters presented and tested as part of this study can be used by ITD's district traffic engineers to develop snow/ice signal times plans for isolated intersections throughout the state.
- The snow/ice signal timing plans can be implemented during snow/ice conditions manually or through the deployment of the weather responsive system developed and tested as part of this study.
- The weather responsive system presented in this study is ready for implementation at the local intersection level. More research and development are needed before a state-wide weather responsive system can be implemented at the state level.
- While the states' RWIS provides a good source of weather data for the proposed weather responsive traffic signal system, using weather data from weather stations that are located at the signalized intersections improve the reliability of the weather responsive traffic signal system. These stations provide weather data that are specific to the signalized intersection and surrounding traffic network locations.

References

1. **Maki, P. J.** 1999. "Adverse Weather Traffic Signal Timing." In: *Transportation Frontiers for the Next Millennium, 69th Annual Meeting of the ITE, Las Vegas, Nevada*, 1999

2. **Al-Kaisy, A. and Z. Freedman.** "Weather-Responsive Signal Timing: Practical Guidelines." *Transportation Research Record: Journal of the Transportation Research Board*, No. 1978, (2006): 49 - 60.

3. **Ahmed, S. M., A. Abdel-Rahim, and M. Dixon**. "An External Logic Processor for NTCIP-Based Traffic Controllers: Proof of Concept for Data Exchange Capability." pp. 779-784 In: *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, Missouri*. New York, NY: IEEE, 2009.

4. **American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE), and National Electrical Manufacturers Association (NEMA).** *NTCIP 1202:2005. NTCIP Object Definitions for Actuated Signal Controller (ASC) Units -– Version 02, a Joint Standard.* Washington, DC: AASHTO, ITE, and NEMA, 2005.

5. Krings, A. "Design for Survivability: A Tradeoff Space", *CSIIRW '08: Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research*. (2008): article 12, New York, NY: ACM.

6. Krings, A. W. "Survivable Systems", Chapter 5, In: Y. Qian, D. Tipper, J. Joshi and P. Krishnamurthy, Editors: *Information Assurance: Dependability and Security in Networked Systems,* San Francisco, CA: Morgan Kaufmann Publishers, 2008.

7. **Lamport, L., R. Shostak and M. Pease.** "The Byzantine Generals Problem." *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3, (July 1982): 382-401.

8. **Thambidurai, P. and Y.-K. Park**. "Interactive Consistency with Multiple Failure Modes," pp. 93-100 In *Proceedings of the 7th Symposium on Reliable Distributed Systems*.New York, NY: IEEE, 1988.

9. **Azadmanesh, M. and R. Kieckhafer**. "Exploiting Omissive Faults in Synchronous Approximate Agreement," *IEEE Transactions on Computers*, Vol. 49, No. 10,, (Oct. 2000): 1031-1042.

10. **Krings, A. W., W. S. Harrison, N. Hanebutte, C. Taylor, and M. McQueen**. "A Two-Layer Approach to Survivability of Networked Computing Systems," pp. 1 – 12 In. *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, 2001.

11. **Munson, J., A. Krings and R. Hiromoto.** "The Architecture of a Reliable Software Monitoring System for Embedded Software Systems," pp. 765 – 774, In: *Proceedings of the 5. International Topical Meeting on Nuclear Plant Instrumentation Controls, and Human Machine Interface Technology,* La Grange Park, IL: American Nuclear Society, 2006.

12. **Krings, A. W. and T. Feyer.** “The Byzantine Agreement Problem: Optimal Early Stopping,” pp. 1-12 In: *Proceedings of the 32nd Hawaii International Conference on System Sciences*, 1999.

13. **Azadmanesh, A., A. W. Krings, and B. Ghahramani.** “Global Convergence in Partially Fully Connected Networks (PFCN) with Limited Relays,” *International Journal of Information Technology and Decision Making (IJITDM),* Vol. 2, No. 2, (June 2003): 265 – 285.

14. Gettman, D., P, Lili, T. Sayed, and S. Shelby. *Surrogate Safety Assessment Model and Validation: Final Report*. McLean, VA: Federal Highway Administration, Turner-Fairbank Highway Research Center, FHWA-HRT-08-051, 2008.

15. Ricks, M. *Potential Safety and Operational Benefits of Weather Responsive Traffic Signal Systems*, Moscow, ID: M.Sc. Thesis, University of Idaho, 2012.