Optimal Communications Systems and Network Design for Cargo Monitoring

Daniel T. Fokum

Submitted to the graduate degree program in
Electrical Engineering & Computer Science and the
Graduate Faculty of the University of Kansas
School of Engineering in partial fulfillment of the
requirements for the degree of Doctor of Philosophy

Dissertation Committee:

Chairperson: Victor S. Frost

David W. Petr

Gary J. Minden

Tyrone S. Duncan

Joseph B. Evans

Date Defended

The Dissertation Committee for Daniel T. Fokum certifies that this is the approved version of the following dissertation:

Optimal Communications Systems and Network Design for Cargo Monitoring

Committee:

Chairperson: Victor S. Frost

David W. Petr

Gary J. Minden

Tyrone S. Duncan

Joseph B. Evans

Date Approved

Abstract

In 2006 the Federal Bureau of Investigation (FBI) estimated that cargo theft cost the US economy between $15 and $30 billion per year. Others have noted that the indirect costs–from investigation and insurance payments–of cargo theft can be two to five times the direct losses from cargo theft. At the same time that the shipping industry is grappling with high cargo theft, exports from Asia to the USA have increased significantly resulting in bottlenecks at certain key ports on the USA's Pacific Coast. Some shipping organizations have sought to get around the bottlenecks at West Coast ports by using inland ports. To this end, they seek to offload cargo from ships directly onto trains destined for an inland intermodal traffic terminal. Once at the terminal, the freight can then be processed by Customs and then distributed within the United States. For such an effort to succeed shippers must have "visibility" into rail shipments.

This dissertation studies the system trade-offs that arise when providing visibility into cargo shipments in motion. This visibility is provided through the optimal placement of sensor and communication technology. This dissertation shows that a transportation security sensor network for monitoring cargo in motion can provide timely event notification to shippers. Two generalized models—one for use when all network elements are on the train and the other for use when some are located trackside—suitable for analyzing a cargo monitoring system are presented. The models show that, under reasonable assumptions, sensor deployment reduces the overall cost of a cargo monitoring system. The models developed here enable system trade-off studies that show that the system deployment cost is inversely related to the deadline for decision maker notification. Furthermore, the system trade-off studies show that the system deployment cost is inversely related to the average train speed. The generalized models developed in this research are Mixed Integer Nonlinear Programs (MINLP). Prior research has shown that MINLP are nondeterministic polynomial time (NP) hard problems. As a result the system trade-off studies are conducted on relatively small trains with 15 units and 33 containers. Thus, a heuristic has been developed to choose the best (or close to best) way to deploy sensors to trains of

arbitrary size. The heuristic has been successfully applied to a train with 105 units and 225 containers.

This dissertation demonstrates that sensors and communications systems can be used to monitor cargo in motion. In addition the dissertation provides potential designers of cargo monitoring systems tools which can be used to study the trade-offs inherent in such a system. Finally, the dissertation presents a heuristic that can be used to deploy sensors to relatively large trains.

Acknowledgments

The completion of this dissertation has only been possible with the support, direction, and guidance of many important people who cannot all be acknowledged in a single page. My heartfelt thanks go to my wife, Yewande, who has walked alongside me during the process of completing this research. Her walk with God, prayers, and encouragement have been a constant encouragement during this journey. I would also like to thank my parents Dr. Gad W. Fokum and Nicole Fokum for their wisdom, support, and their instilling in me the values of honesty, hard work, and persistence.

I would like to thank my dissertation advisor, Dr. Victor Frost for his support of this project, for suggesting lines of research that may not have been immediately apparent to me, for re-reading the many drafts of this dissertation, and for his persistence in insisting on high quality results. I also thank Dr. David Petr, Dr. Gary Minden, Dr. Joseph Evans, and Dr. Tyrone Duncan for their participation on my dissertation committee. I am also grateful to the many teachers and fellow students who have helped expand my knowledge and understanding of technical subjects and writing.

Special thankfulness is also extended to Oak Ridge National Laboratory (ORNL) for supporting portions of this work under Award Number 4000043403.

As I complete this work I am reminded of the truth of this verse from Psalm 127:1[1], "Unless the LORD builds the house, its builders labor in vain. Unless the LORD watches over the city, the watchmen stand guard in vain." I dedicate this work to the triune God, the source and giver of life, who loved me and gave Himself up for me.

[1]Scripture taken from the Holy Bible, NEW INTERNATIONAL VERSION®, Copyright©1973, 1978, 1984 by Biblica, Inc. All rights reserved worldwide. Used by permission.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2006 the FBI estimated that cargo theft cost the US economy between $15 and $30 billion per year [1]. Mayhew [2] noted that the indirect costs–from investigation and insurance payments–of cargo theft can be two to five times the direct losses from cargo theft. Cargo theft affects originators, shippers, and receivers as follows: originators and receivers need a reliable supply chain to deliver goods in a timely and cost-effective manner. Shippers hold liability and insurance costs for shipments, which are proportional to the rate of theft. Finally, receivers are impacted by out-of-stock and scheduling issues due to cargo theft.

At the same time that the shipping industry is grappling with high cargo theft, exports from Asia to the USA have increased significantly resulting in bottlenecks at certain key ports on the USA's Pacific Coast. Some freight transportation organizations have sought to get around the bottlenecks at West Coast ports by using inland ports. To this end, they seek to offload cargo from ships directly onto trains destined for an inland intermodal traffic terminal. Once at the terminal, the freight can be processed by Customs and then distributed within the United States. Given the risk of cargo theft, secure trade lanes are needed between ports and intermodal traffic terminals. This work explores the system trade-offs when designing communications systems and networks for monitoring cargo in motion.

1.1 Motivation

For the success of the scheme described above, shippers need to be able to gain "visibility" into freight and cargo movement, particularly in intermodal 'black holes," where freight changes hands across modes and carriers. Visibility will only be possible through real-time integration of sensor data with carrier, shipper, broker, importer, exporter, and forwarder information. Unfortunately, different complex systems are currently used in the container transport chain [3].

To achieve the objective of providing visibility into cargo shipments, trains, railcars, and containers will be equipped with sensors and devices that communicate sensor status, sensor ID, and train location. Breaking a sensor on a container would generate a message that is communicated to a reader over a network and then in near real-time to train personnel and/or to an operations center as an alarm message. Sensor readers may be distributed along the train in such a manner that all the sensors would be connected to one or more readers on the train. In addition, location information will be sent with the alarm so that the geographic location of the breakage event can be identified. Shipment information from a Trade Data Exchange (TDE) [4] will be included in the alarm so that the rail car, container, and its contents can be identified. While sensors will present a non-negligible initial cost, their use could allow the sensing system to demonstrate shipment integrity. It is also expected that the use of such systems may help reduce the risk of cargo theft.

The objective of the research presented here is to develop models to find the "best" system design including communications network and locations for sensors in a rail-based sensor network, as well as to guide the design of future cargo monitoring systems. These models will also be applied to determine system trade-offs when monitoring cargo in motion.

1.2 Problem Statement

In this section we introduce a generalized problem statement. In this dissertation we seek to deploy electronic devices such as sensors, repeaters and radio networks associated with the train infrastructure to provide cargo visibility. We may be able to achieve visibility into a cargo shipment on a train by placing sensors, readers, and backhaul communication devices on every container on a train (as is done today for high-value cargo [5] or hazardous material), or by deploying sensors on every container on the train and closely placing readers with backhaul communications capabilities at the trackside. However, the cost and system trade-offs for such approaches are unknown. As a result this research is aimed at answering the following system design question:

> Given a collection of containers and a collection of end-to-end information subsystems (including sensors, seals, readers, and networks); how do we design an end-to-end system that meets the "visibility" constraints for all containers while minimizing overall system cost?

In Chapter 4 we provide a rigorous definition of visibility. In our specific rail scenario, our overall design question spawns the following issues:

1. How to map and analyze a "system" description of containers on railcars, train scenario—including train speed and trips per time unit—and associated communications infrastructure into the visibility space? Thus an appropriate system model needs to be developed.

2. How to assign a cost to every position in the visibility space?

3. How to use 1) and 2) to find minimum "cost" systems for providing visibility into a rail shipment?

4. How to use 1) and 2) to determine system trade-offs when seeking visibility into rail shipments?

The research described here answers these questions.

1.3 Results Summary and Contributions

This section provides a summary of the results from our work along with a brief discussion of our societal contributions.

1.3.1 Results Summary

At the conclusion of this study we will produce models that return the optimal assignment of sensors to containers on a train and the associated system cost metric when given the train scenario and sensor specifications. These models will also be used to study system trade-offs, such as:

- A trade-off between a train-mounted and a trackside deployment of readers.
- A trade-off between system cost and the time needed to report events.
- A trade-off between the number of sensors assigned and the value of the goods on the train.

Furthermore, the models will be used to determine the sensitivity of the system cost to various system parameters. The models developed here are Mixed Integer Nonlinear Programs (MINLP), which cannot be solved for trains of realistic scale due to computational limitations. A heuristic has been developed for these cases to generate a system design that is as close to the best as is possible for a given train scenario.

1.3.2 Contributions

It has been observed that the deployment of Intelligent Transportation Systems (ITS), such as electronic seals and readers to monitor cargo, can lead to a decrease in cargo insurance premiums since pilferage and cargo loss are reduced [6, 7]. Moreover, ITS technologies could increase customer confidence in the transportation chain as well as lead to increased efficiency in shipments. Finally, we expect that the use of ITS could also result in better homeland security by reducing the risk of contraband insertion into shipments. Our work resulted in cost-effective system designs that enable shippers to demonstrate to insurers that thefts can be detected with a certain probability and communicated to decision makers within a specific timeframe with known probability. The societal contributions of this study include:

- The analysis of data from a field trial of a cargo monitoring system to show that commercial-off-the-shelf devices can be used for timely notification of decision makers of events on a train.

- The formal definition of visibility for cargo monitoring systems.

- The development of a mathematical model, including a railcar, container, and sensor indexing scheme, that leads to cost-effective practical system design for cargo monitoring.

- The study of system trade-offs when designing systems for monitoring cargo in motion. The trade-off studies show that the system deployment cost is inversely related to the deadline for decision maker notification. Furthermore, the system trade-off studies also show that the system deployment cost is also inversely related to the average train speed.

- A heuristic to aid in the design of systems of realistic scale for monitoring cargo in motion.

1.4 Document Outline

The rest of this dissertation is laid out as follows: A review of different disciplines that are relevant to this study is presented in Chapter 2. Our review of literature includes fields such as intermodal transportation, optimization theory, communications aboard trains, and intrusion detection. Chapter 3 presents an open system transportation security sensor network for monitoring cargo and presents experiences from a sensor network field trial. In Chapter 4 we present two models for analysis and design of communications systems for monitoring cargo in motion. One of the models is intended for use when the sensors and communications infrastructure is on the train, while the second model is intended for the case where the sensors are on the train and the readers are at the trackside. Some of the results from Chapter 3 are used as inputs to the models presented in Chapter 4. Chapter 5 uses the models from Chapter 4 to study the system trade-offs that exist when designing communications systems and networks for monitoring cargo in motion. The results from Chapter 5 will provide customers with tools with which to balance system performance and cost. Our experience showed that the optimization models could not be used easily on trains that had more than 33 cars; thus, a heuristic was developed to aid in the design of communications systems and networks for larger trains. Chapter 6 presents the heuristic and compares its performance with the optimization approach. Finally, we provide some concluding remarks in Chapter 7.

Chapter 2

Review of Literature

In this chapter we review some of the relevant literature. In Section 2.1 we provide an overview of intermodal transportation with a focus on containers and container security as well as trains and rail yard operations. In Section 2.2 we give a very brief overview of optimization theory, while Section 2.3 discusses how optimization theory has been applied to train operations. Section 2.4 presents some of the technologies that have enabled monitoring of freight on trains, such as sensor networks, intrusion detection, broadband Internet access on trains, and software based on a service-oriented architecture. Section 2.5 concludes this chapter by highlighting common themes that emerged in the literature review.

2.1 Intermodal Transportation

Intermodal transportation may be defined as "the movement of goods in one and the same loading unit or road vehicle, which uses successively two or more modes of transport without handling the goods themselves in changing modes [3, ch. 1]." In the case of rail transportation this transfer between modes of transport is done either by circus loading (using ramps between rail cars to form a temporary roadway upon which cargo may be moved), gantry loading (using a large overhead crane to move cargo from roadway to train), or side loading (sliding cargo from road chassis to flat car) [8, ch. 15].

Intermodal transportation has greatly benefited from the introduction of the double-stack container car. In fact, by 1994 container loadings onto trains constituted 53% of the total intermodal traffic borne by trains [8, ch. 15]. In addition, by making use of intermodal railcars, train companies are able to use their fleets more efficiently thereby generating more revenue. Armstrong [8, ch. 15] notes that the fleet-wide average number of revenue loads for intermodal railcars is roughly about three times that achieved by other freight cars.

In the next two subsections we examine two components of intermodal transportation with respect to rail transport. We examine trains and rail yard operations in one subsection, and then examine containers and container security in another.

2.1.1 Trains and Rail Yard Operations

Armstrong [8, ch. 12] provides an overview of train operations. Typically trains are formed by assembling cars from various sources into "blocks" that are headed to a common destination. The blocks are then strung together into a train. In general, cars are set out in station order on the train, with cars for the first station at the head end, and cars for the next station in the following block, etc. Full details on how blocks are formed are found in Armstrong [8, ch. 12].

The use of inland ports to distribute cargo is facilitated by unit trains. Unit trains travel between two given points, and they have proliferated in North America due to a change in the regulatory climate in the United States and Canada. As a result of these regulatory changes shippers are now able to sign contracts with train companies that permit a single train and group of locomotives to be assigned to their service. This results in a high degree of utilization for rail equipment as well as benefits for both the shipping company and the train company. As of 1998, unit trains were

heavily used for shipping coal, containers across land, semi-finished steel, crude oil and many other products [8, ch. 14].

2.1.2 Containers and Container Security

Most of world's non-bulk cargo travels in marine shipping containers, which are typically reinforced steel boxes with one double door for access on one side. These containers generally vary in length from 10–62 feet (3.0–18.9 m), and come in a wide assortment of types ranging from refrigerated boxes to full-tank containers to "dry boxes" [3, ch. 2].

Shipping containers are heavily used in intermodal transportation so they also present a security vulnerability. As a result the European Conference of Ministers of Transport (ECMT) [3] makes the following observations and recommendations regarding container security:

- The focus for container security should be providing information at the right time, not real-time tracking. However, certain government agencies might require real-time tracking.

- The specific container stuffing location is important to container security, since it represents the last point where the container contents may be inspected and reconciled with the bill of lading and/or invoice [3, ch. 2].

- Containers are most vulnerable to being tampered with when they are at rest, and least vulnerable when they are in motion [3, ch. 2].

- Land-side border crossings represent a particular vulnerability with respect to container security, since they may sometimes have poor customs control and

inefficient processes that create delays that can be exploited by persons who seek to tamper with containers [3, ch. 2].

- A comprehensive risk assessment should be carried out for container security. Amongst other things, this risk assessment should analyze strategic threats for container security and develop appropriate responses [3, ch. 3].

- The physical integrity of the container must be assured as the container moves from origin to destination. Finally, the security of the container environment must be ensured as the container is in transit and being handled in the container transport chain [3, ch. 4].

Intermodal shipping containers can be secured by using electronic "seal" technology such as active RFID tags that can indicate if and when a container door was opened. Other sensor technology can also be used to enhance container security. In 2005 Fuhr and Lau [9] stated that mesh networking could be used to link wireless sensors inside containers. Schoeneman *et al.* [10] used sensors and GPS receivers for tracking and monitoring radioactive items and individual containers during shipment. Their tracking system combines location information from GPS receivers with sensor data, encrypts and transmits the combined data using the INMARSAT system in near real-time, e.g., 5 minutes.

The sensors that will be used to secure container shipments will most likely be resource-limited devices. Furthermore, data needs to be secured while in transit between the sensor on the container and the operations center. In 2007 Lauf and Sauff [11] proposed a security protocol for transmitting information from sensors within a shipping container to a trusted third party. Such a protocol permits tracing liability for cargo theft and/or damage while minimizing the risk that shipping containers can

be used for terrorism or shipment of contraband. The architecture used in [11] consists of sensors, sensor nodes, and a MASC (Monitoring and Security of Containers) unit. The sensors collect data from the environment, while sensor nodes possess a processor and host several sensors. Finally, the MASC unit collects information coming from external sensors, and processes it.

In [11] it is observed that it is cheap to ship containers by ship; however, there is also a low quality of service associated with this mode of transportation, since containers may be poorly handled. Thus, it is desirable to trace liability for cargo theft and/or damage using sensor technology. Sensor technology can also minimize the risk that shipping containers can be used for terrorism or shipment of contraband. In order to prevent the insertion of bogus data into the communications between sensors and an operations center, a security protocol must be developed to secure communications between sensors and a remote database. In this instance the security protocol will be deployed to the MASC units. It is expected that the MASC units will be long-lived battery-powered devices that need to operate for several years while supporting lightweight communication protocols. The MASC unit will connect to a remote database server to report sensor information. A symmetric key is shared between the remote database server and the MASC unit to allow for confidential communications between the remote server and the MASC unit. In [11] the MASC unit communication protocol was deployed to some test hardware, and experiments show that more RAM was needed on the test hardware. Reference [11] concludes by observing that additional work is needed to create tamper-resistant MASC units.

Decker *et al.* [12] also envision the presence of "Smart Items" in shipments. Smart Items are enabled by placing either barcodes, RFID tags, or sensor nodes on goods to enable the identification, tracing, location tracking, and monitoring of cargo. Decker

et al. [12] develop models to estimate the profit resulting from the deployment of smart items for suppliers, shippers, and the final customer. A simple use case is also developed and it is shown that it is advantageous to deploy powerful smart items, such as sensors [12].

2.2 Optimization Theory

Thus far we have covered intermodal transportation and some of the security issues that arise with intermodal transportation. In this section we introduce optimization theory and briefly discuss how it has been applied to train traffic.

In engineering, science or finance disciplines we are frequently concerned with finding the best solution to a given problem. The process of finding the best solution to a problem is optimization, and this process is needed, for example, to find the cheapest way construct a computer network. In order to apply optimization techniques, one needs to identify an objective function, which is a quantitative description of the system being improved, e.g., the objective function can map the description of cargo monitoring system to a cost. The objective function of any system depends on certain system characteristics, variables, which can be mapped to numerical values [13]. Quite frequently these variables are subjected to certain constraints that require the objective function to model reality more accurately. For example, we can require the transmission range of a given sensor to exceed a certain length so that a communications network can be formed. A goal of this research is to analyze an objective function and constraints for optimal placement of containers and sensors on a train. Having formed the objective function and its constraints, the goal of every optimization problem is to find the right combination of variable values such that the objective function is minimized, while the constraints are not violated [13].

Optimization problems can be grouped into two main classes based on whether their variables are discrete or continuous. Problems that have discrete variables are called combinatorial optimization problems [14].

More formally an instance of an optimization problem may be defined as follows:

Definition 1 (Papadimitriou and Steiglitz [14])**.** *An instance of an optimization problem is a pair (F, c), where F is any set, the domain of feasible points; c is the cost [objective] function, a mapping*

$$c : F \to \mathbb{R}^1$$

The problem is to find an $f \in F$ for which

$$c(f) \leq c(y) \quad \forall\, y \in F$$

Such a point f is called a globally optimal solution to the given instance, or when confusion can arise, simply an optimal solution.

In general optimization problems will take the form [15]:

$$\begin{aligned} &\text{minimize } f_o(x) \\ &\text{subject to } f_i(x) \leq b_i, \qquad\qquad i = 1, \ldots, m \end{aligned}$$

The vector $x = (x_1, x_2, \ldots, x_n)$ contains all the variables for the optimization problem, while $f_o : \mathbb{R}^n \to \mathbb{R}$ is the objective function for this problem. The functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are the constraint functions. The intent here is not to provide an in-depth treatment of optimization theory. For a more complete treatment please consult [13–15]. In Sections 2.2.1 and 2.2.2 we provide a brief review of mixed inte-

ger linear programming and mixed integer nonlinearly constrained programming; two classes of optimization problems which appear in this dissertation. In Section 2.3 we provide a very brief overview showing how optimization theory has been applied to container transshipment yards and to optimization of train traffic.

2.2.1 Mixed Integer Linear Programming

A mixed integer linear program (MILP) is an optimization problem where the objective function and all of the constraints are linear functions, while some of the variables are integer-constrained [16]. Darby-Dowman and Wilson [17] state that integer program models are generally harder to solve than linear program models of the same size, while Bonami *et al.* [18] state that MILP are $\mathcal{NP}$ (nondeterministic polynomial time) hard problems. Mixed integer linear programs are either solved by branch-and-bound, branch-and-cut, or branch-and-price methods.

When solving integer programs a tree of the entire solution space is created, the root node of the tree is the entire state space, S, while all other nodes represent smaller partitions of the solution space. With branch-and-bound the branching is done by selecting a variable x with a fractional value k and then creating two sub-problems with the additional constraints $x \leq k$ and the other $x \geq k+1$. This is called the Linear Programming Relaxation (LPR). At a selected node of the tree the integer program LPR is solved. If there is no feasible solution to the problem at that node, the node is eliminated. Otherwise if the solution of the linear programming relaxation is integer feasible and the objective function solution is less than the previous upper bound then the objective function value for this subproblem is set as the new upper bound for the objective function. Branching continues until the best integer feasible solution found is shown to be optimal. With branch-and-cut at each stage in the development of the

solution space tree an equation called the cut is added to the set of constraints when carrying out the linear programming relaxation. The cut has the added requirement that it must not exclude any integer solutions at that node or any of its descendants; however, it may exclude integer solutions for preceding nodes. With branch-and-price an auxiliary problem is solved to identify which columns should be added to the linear programming relaxation. The relaxation is optimized and more columns are identified for addition to the LPR [17].

2.2.2 Mixed Integer Nonlinear Programming

A mixed integer nonlinear program (MINLP) is an optimization problem with some integer-constrained and continuous variables as well as nonlinear constraints and/or objective function. If all the variables are continuous, then we have a nonlinear program. MINLPs are a superset of mixed integer linear programs, where the reduction to MILP takes place when all of the functions in the optimization problem are linear [16].

Mixed integer nonlinear programs are $\mathcal{NP}$-hard [18]. However convex MINLPs can be solved using the following techniques: branch-and-bound, extended cutting plane, outer approximation, generalized Benders decomposition, LP/NLP-based branch-and-bound, and branch-and-cut [16, 18]. This section provides an overview of each of these techniques. More detailed explanations of the solution methods are found in [18]. Branch-and-bound for MINLPs is done just as for mixed integer linear programs, except that a nonlinear program is now solved at each node of the tree [16]. The extended cutting plane method constructs a mixed integer linear program relaxation and solves it. If the solution is not feasible, then a cutting plane of the most violated constraint at the optimal solution is added to the relaxation and the

problem is re-solved and the process is repeated [16, 18]. Outer approximation (OA) is based on the observation that a MINLP is equivalent to a MILP of finite size. The MILP can be generated by linearizing both the objective and constraint functions. The linearized function is then solved and the integer solution from this step is used as a bound on the optimal value of the NLP. This process is repeated until the upper and lower bounds of the optimal value of the nonlinear program are within a specified tolerance [18]. Generalized Benders decomposition is very similar to the outer approximation method except that it has only one continuous variable [16]. LP/NLP-based branch-and-bound is an extension of the OA method. It uses LPR to find an integer solution in a branch-and-bound tree and then solves the nonlinear program to get upper bounds on the solution [16, 18]. Branch-and-cut has been adapted to solving MINLPs [16]. This method is similar to branch-and-bound, but it adds cutting planes at each node of the tree to strengthen the NLP relaxation [16].

2.3 Application of Optimization to Trains

Train operations lend themselves readily to optimization techniques provided good models can be constructed to describe the system that is being optimized. Optimization techniques have been used for a wide variety of applications ranging from improving aerodynamic efficiency of trains [19], determining optimal sites for placement of defect detection equipment [20], train routing and scheduling [21–24], reducing the amount of time used to process trains in intermodal rail yards [25, 26], as well as for providing broadband Internet access to users on fast moving trains [27].

2.4 Enabling Technologies

Freight monitoring is possible because of technologies such as sensor networks, intrusion detection and perimeter security, and the ability to provide Internet access on trains. In the next three subsections we provide very brief introductions to these technologies. This section also provides an overview of service-oriented architecture software, since this has already been tested in rail environments for tracking cargo and for frequently serviced parts.

2.4.1 Sensor Networks

Sensor networks are an emerging application of advanced wireless networking and computer technology. Sensor networks have been used to monitor railway equipment [28, 29] and have been proposed for use in monitoring cargo in motion [12, 30, 31]. Sensor networks typically consist of a set of small resource-constrained computers, called sensor nodes that collect data from their environments and then transmit that data to a base station, or other central site. In general a wireless sensor node (WSN) consists of a sensing device, e.g., an electronic nose, a temperature sensor or a motion detector, a small microprocessor, a radio and a limited energy source. It should be noted that when a sensor node is connected to just one sensor, the sensor node is sometimes called a sensor, which causes some confusion [32]. Base stations, unlike wireless sensor nodes, generally have radios, but will have available more computing resources and a larger energy source. The base stations generally aggregate information from the nodes and then pass them on to other computers for presentation [32]. Sensor networks are covered in much greater detail in [33]. This section presents material on surveillance wireless sensor networks, a class of sensor networks that was not covered in [33].

In 2003 Avancha *et al.* [34] argued that there is a need to identify problems and challenges to sensor networks, so that good solution techniques can be applied to address these challenges. Reference [34] examines a sensor network that can be used to provide a security perimeter around an individual. Some of the problems that can be expected in this class of sensor networks include:

- passive information gathering,
- subversion of a sensor node,
- insertion of a false node,
- dealing with the addition of a legitimate node.

In developing a methodology to address these problems, [34] assumes that the sensor network's base station is computationally robust, key management and re-keying mechanisms are not needed at base station, the communication paradigm is one-to-one, and the physical security protocol becomes more specific as one approaches the base station. The security requirements for this sensor network include authentication, integrity, confidentiality, anti-playback and resistance to traffic analysis. In this sensor network it is also assumed that the base station is deployed with a unique symmetric encryption key and a key that it shares with every node in the sensor network. Sensor nodes are assumed to be deployed with a unique symmetric key that is shared with the base station, as well as the base station's key. Messages destined for the base station are encrypted with the base station's key as well as the symmetric key that is shared by each node and each base station. In Avancha *et al.*'s [34] scheme message integrity is achieved through the use of an encryption algorithm, while anti-playback is achieved by use of a timestamp. Finally, message privacy is guaranteed by encrypting all communications. New nodes to the sensor network can

be accommodated simply by adding the new nodes' keys to the base station. The new nodes would also be supplied with the base station's key. Reference [34] concludes by presenting simulation results that show that energy consumption for topology discovery and network setup decreases as the ratio of adjacent nodes to non-adjacent nodes increases.

In a surveillance wireless sensor network (SWSN) breach probability is defined as the probability of missing an unauthorized target that traverses the sensor network [35]. The probability of a target traversing the sensor network across the breach path gives a notion to the level of security offered by the SWSN. Onur *et al.* [35,36] present research on the number of sensors needed to achieve an acceptable breach probability in a SWSN. Some of the research questions covered in [36] include:

- How many sensors should be deployed to achieve a required security level?
- How should sensor detection be modeled and sensing coverage determined?
- What are the effects of the geographic properties of the field on target detection?
- How should sensors be deployed?
- How can breach paths be discovered?
- How could false alarms be minimized and decisions about target detection be improved with collaboration? What is the impact of sensor scheduling on sensing coverage?

Some of the questions listed above are similar to our questions, for example, we seek the number of sensors needed to achieve a required security level, how should sensor detection be modeled, and how should these sensors be deployed. On the other hand,

our work does not address how breach paths may be discovered nor do we examine how target detection may be improved with collaboration.

In [36] false alarms in the SWSN can be categorized as either transient or persistent. Transient false alarms can be removed with an exponentially weighted moving average, while persistent false alarms can be removed via in-network aggregation of data.

The breach path in the network can be found by locating the path through the network that is as far as possible from sensor nodes, i.e., find a path between two points where the sum of the closest distance to the sensor nodes is maximized. One can then find effective locations for the sensor nodes by splitting the coverage area into grids and then maximizing the average coverage of the grids, or maximizing the coverage of the least effectively covered grid. If the sensing field is defined as a cross-connected grid, detection probabilities can then be computed for each grid point using sensor models[1]. A target seeking the breach path through the SWSN will try to traverse the network by starting at a point on the insecure side of the network to get to the opposite side of the network. The weakest breach path problem may then be seen as finding the permutation of a set of grid points so that a target can traverse the sensor network from the starting point to the destination point with the least probability of detection. We can then find the weakest breach path by maximizing the probability of missing a detection along a given path, where the breach detection probability is defined as the maximum probability of detection along the breach path.

Simulation results show that when sensors are deployed deterministically, the required number of sensors is smaller than in random deployment schemes. Onur *et al.*

[1]These models include a Neyman-Pearson detector that maximizes probability of detection subject to a fixed false alarm rate, and Elfes's model for sensors that defines the probability that sensor k detects an event as an exponentially distributed random variable. In Elfes's model, if the distance between the sensor and the target is less than some value, the target is always detected.

[36] conclude by noting that breach detection probability is most sensitive to false alarm rate, so sensors need to be placed such that they have an unobstructed view of target. In addition, increasing the amount of data collected for each breach decision by a sensor enhances the overall performance of the sensor network.

Sensor fusion is sometimes used to combine data from multiple sensors and nodes in order to get a higher probability of detection and a lower probability of false alarm [37]. In 2006 Krakow *et al.* [37] used a Partially Observable Markov Decision Process (POMDP) to control a surveillance wireless sensor network (called a perimeter security system in [37]). According to [37] one challenge in a sensor network is to maximize the information collected in a sensor network while minimizing some cost metric (for example power consumption). The Partially Observable Markov Decision Process (POMDP) model is used to trade-off short-term benefits in sensor network performance with improved performance over the long run. Given a set of M sensors distributed to track T targets, the goal is to select the number and combination of sensors to trade-off tracking accuracy and sensor usage. The POMDP can be solved approximately to know which sensors need to be activated for improved sensor network performance [37]. The POMDP sensor scheduling algorithm has also been deployed to a wireless sensor network testbed, but no testbed results have been reported.

Hitherto, we have only examined the role of sensor networks in wireless surveillance, but other roles exist for sensor networks. For example, they can be used for monitoring food in the transport chain. Shan *et al.* [30] present a review of sensor networks designed for refrigerated delivery vehicles. An intelligent sensor is defined as one that can change its behavior to improve its data collection ability [30]. This behavior includes self-calibration, self-validation, and self-compensation. (Self-

calibration is done when a sensor would like to recover its performance. Self-validation uses mathematical modeling techniques to describe sensor faults. Self-compensation uses artificial intelligence models to achieve high sensor accuracy.) Shan *et al.* [30] conclude by suggesting that artificial intelligence techniques such as neural networks and fuzzy logic will allow for intelligent sensors that can perform self-calibration, self-compensation, and self-validation.

In 2007 Ruiz-Garcia *et al.* [31] provided a survey on the use of sensor networks to monitor fruit transportation in intermodal, refrigerated containers. Ruiz-Garcia *et al.* [31] state that the use of sensor networks in the supply chain can allow for traceability of shipments, with accurate information being provided on the products involved in the chain. Traceability can be done by using RFID technologies to "seal" containers electronically. The seals may be written and read by handheld or trackside readers, while the RFID tags would have memory that can record the date and time of all events seen by the tag. Reference [31] concludes by noting that the technology now exists to allow for the development of a monitoring system for refrigerated containers. Such a monitoring system should include different types of sensors in various locations of the container. Sensor readings and GPS information can be combined to track shipping containers through different stages of the supply chain [31].

In this research we examine the optimal placement of sensors on a train for cargo monitoring. Previous work [38] shows that placement of sensors is, in general, an $\mathcal{NP}$-hard problem; however, approximation algorithms exist for sensor placement. Chakrabarty *et al.* [39] examine the placement of different types of sensors in a field such that a desired amount of coverage is achieved, and cost is minimized. Reference [39] uses an Integer Linear Program (ILP) to solve this problem. They find that as the number of sensors required to "cover" each grid point increases, it is more economical

to use more expensive sensors that have a longer sensing range. Furthermore, exact solutions to the ILP take longer to solve as the problem size increases; therefore, they opt for a heuristic approach to the problem.

Efrat *et al.* [38] present two approximation algorithms for sensor networks. The first algorithm shows how to place base stations in the sensor network such that the network's lifespan is within some bound of the optimal lifespan[2]. The second algorithm shows how to minimize the number of sensors placed throughout the network so that at least two sensors "cover" each point from substantially different directions. No empirical results are presented in [38]; however, proofs are used to show the time complexity of the schemes, as well as to set bounds on the algorithms' results.

2.4.2 Intrusion Detection and Perimeter Security

To computer scientists and electrical engineers, mention of intrusion detection frequently conjures up images of detecting unauthorized access to computer resources. In this dissertation we are also interested in intrusion detection, but this topic will refer to detecting unauthorized physical access to containers or shipyards. In this respect, some of the material on surveillance wireless sensor networks that we presented in Section 2.4.1 is applicable here; however, in this section we go beyond surveillance wireless sensor networks to highlight some of the technologies that are used to protect perimeters.

One of the earliest, and interesting approaches, for intrusion detection used disturbances to commercial FM radio signals caused by an intruder to alert operators to a breach [40]. Thermal imaging systems can also be used for intrusion detection, as shown in [41]. Bisbee and Pritchard [41] provide a review of state-of-the-art ther-

[2]Here lifespan is defined as the length of time until the first sensor exhausts its battery supply.

mal imaging systems in 1997. They concluded that thermal imaging systems allow operators to make threat assessments while maintaining confidence in the intrusion detection system. However, intrusion detection systems were in need of further improvement.

Given a set of sensors out in the field, intrusion detection may also be done by using computers to filter out false alarms from the set of alarms received. Horner *et al.* [42] present the AutoMatic Event auTHentication SYSTem (AMETHYST), which automatically scans data from cameras to provide outdoor intrusion detection. Computers are used to analyze closed circuit television images just before and after an alarm is detected. The system sends an alarm to a human operator if there is insufficient data to conclude the cause for an alarm.

In 2000 Bass [43] discussed the use of multisensor data for doing network intrusion detection. While [43] deals with network intrusion detection, some of its conclusions are relevant to physical intrusion detection consequently Bass's key findings are summarized here. Bass [43] suggests, as others have, that data from multiple sensors and sources should be used to infer information about events, activities, and situations. The decision support systems for situational awareness, should operate on the *Observe-Orient-Decide-Act* model. *Observe* tasks include collection of data by sensors, *orient* tasks involve using data mining to discover or learn previously unknown facts from the data collected, *decision* tasks result in information being refined to identify threats. Finally, *Act* tasks are the responses to any observed attacks [43].

Another approach for perimeter security is to use mathematical analysis to determine the vulnerability of structures to threats. Tarr [44] discusses a mathematical model that allows for the evaluation and comparison of performance and costs of a set of alternative security perimeters.

Jacobson *et al.* [45] also apply analysis to study the aviation security system. They state that an effective security system should minimize its false alarm rate while holding the missed detection rate fixed. They then develop a system response function that "combines the responses of all devices in an access control security system" and returns a result that can be translated into either an alarm or a clear. Using the system response function, the False Alarm, False Clear (FAFC) problem is formulated to minimize the false alarm probability of a system while holding the probability of a false clear subject to an upper bound. Jacobson *et al.* [45] then show that the False Alarm, False Clear (FAFC) problem is $\mathcal{NP}$-complete. However, it can be solved in polynomial time if all the system responses given a threat condition and all the system responses given a non-threat condition can be bounded. In this case, the Greedy and Dynamic Programming heuristic algorithms used to solve the Knapsack Problem can be applied to the FAFC problem [45].

In 2008 a model was developed to minimize construction-related security breaches for airport construction projects while simultaneously minimizing site layout costs [46]. This multiobjective optimization model can be used to comply with FAA guidelines for airport security. Some of the variables in the optimization problem include:

- Security response distance (which is the distance separating each secure facility from the construction site. This variable affects the level of controlling construction-related security breaches and it also has an indirect role on the cost of the site layout plan.).

- Security systems utilized (which control and minimize security breaches. These systems, including security fences and access control technologies, allow for site layout costs to be kept to a minimum.)

- Temporary facilities locations (these locations are based on the available space

defined by the site fence.)

The optimization model uses two criteria to evaluate the impact of site layout decisions. These criteria include the security response distance criterion (SRDC), which specifies buffer zones around each secure facility, and the security systems criterion (SSC), which evaluates the effects of installing security systems in airport construction site layouts. A genetic algorithm is used to solve the optimization problem, and the results show optimal locations for temporary facilities, optimal locations of security fences, and optimal utilization of security control systems [46].

2.4.3 Communications Aboard Trains

In order for cargo to be effectively monitored on trains, there has to be an Internet link between the operations center and the train. A literature survey found in [47,48] shows that we can subdivide networks for providing broadband Internet access to trains into the train-based network, the access network—for connecting the train to the service provider(s)—and the aggregation network—for collecting user packets generated in the access network for transmission to the Internet. Furthermore, the review shows that the current trend is to provide Internet access to passengers on trains using IEEE 802.11; however, a clear method for connecting trains to the global Internet has yet to emerge.

Freight trains can also be monitored to collect operational data from the train. In 2005 Edwards *et al.* [29] presented a prototype system to monitor and control various sensors and actuators on a freight train. The prototype uses a Controller Area Network (CAN) bus to collect data from the sensors. The data is then coupled with GPS information and reported to a web server via a CDMA-based transmitter. Edwards *et al.* [29] argue that "on board sensing of mechanical defects enables car

owners to track defects and proactively schedule maintenance" at a time and location that makes economic sense. In the context of this project we expect that trains would be connected to the Internet either using a GPRS/HSDPA link or a satellite link.

2.4.4 Service-Oriented Architecture Software

Software based on a service-oriented architecture has been tested in rail environments for tracking cargo [49,50] and for monitoring frequently serviced parts [51]. As a result this section provides a very brief overview of service-oriented architectures, while Chapter 3 describes field trial experiences from a transportation security sensor network based on a service-oriented architecture..

When software is designed using a service-oriented architecture (SOA), the components of a system are defined as services and the users of those services are seen as clients. Furthermore, business logic is implemented in the services and communication between the services is based on open standards [52]. A service-oriented architecture allows for the sharing and processing of information in a uniform manner, thereby improving process efficiency.

The Transf-ID system [51], proposed in 2009, uses radio frequency identification (RFID) tags and a service-oriented architecture to track cargo, railcars, and frequently serviced parts. Fernandez *et al.* [51] argue that use of the Transf-ID system improves rail freight safety since part maintenance schedules are now based on actual use.

2.5 Common Themes

In this chapter we covered different topics relevant to our problem space. We gave a brief overview of intermodal transportation and train operations, optimization theory and its application to transportation problems, enabling technologies such as

surveillance wireless sensor networks and communications to trains, and perimeter security. Some of the significant themes that emerged from our literature review include:

- Intermodal shipping containers are least vulnerable when they are in motion.
- Optimization theory is being used for improving train operations.
- Wireless sensor networks are increasingly used for perimeter security. When wireless sensor networks are used for perimeter security we can improve their performance by aggregating data from several sensors. Of particular significance in our review of surveillance wireless sensor networks are Onur *et al.*'s work [35, 36] to locate the weakest breach path and Krakow *et al.*'s work [37] to schedule sensors. In our opinion references [35, 36] are not immediately applicable to our domain because in our problem space sensors are attached to objects with different values, and these values need to be factored into the system model. In addition [36] deals with an intruder who seeks to traverse a surveillance wireless sensor network (SWSN), whereas in our case we assume that intruders are more interested in stealing from the sensors' environment. Krakow *et al.*'s [37] work, on the other hand, might be applicable to our environment.
- Mathematical analysis has been applied to improving perimeter security of facilities.
- Train equipment has been monitored using software built on a service-oriented architecture. In addition, security protocols have been developed for communicating with sensors inside shipping containers. However, our review of literature does not indicate that anyone has combined sensors and an open service-oriented architecture to monitor freight in motion.

Given these findings, in Chapter 3 we present field trial experiences from an open system transportation security sensor network designed to monitor cargo in motion. The data from these field trials will be used to in the models develop in Chapter 4 to find optimal locations for sensors and enable the study of system trade-offs when seeking visibility into cargo shipments.

Chapter 3

An Open System Transportation Security Sensor Network: Field Trial Experiences

3.0 Chapter Summary

Cargo shipments are subject to hijack, theft, or tampering. Furthermore, cargo shipments are at risk of being used to transport contraband, potentially resulting in fines to shippers. The Transportation Security Sensor Network (TSSN), which is based on open software systems and Service Oriented Architecture (SOA) principles, has been developed to mitigate these risks. Using commercial off-the-shelf (COTS) hardware, the TSSN is able to detect events and report those relevant to appropriate decision makers. However, field testing is required to validate the system architecture as well as to determine if the system can provide timely event notification. Field experiments were conducted to assess the TSSN's suitability for monitoring rail-borne cargo. The field trials allow system designers to get experience with the TSSN in a real rail environment. In addition, the results from the field experiments will be used in models to characterize rail-based cargo monitoring systems.

Log files were collected from the field trials and post processed. This chapter presents the TSSN architecture and results of field experiments, including the time taken to report events using the TSSN as well as the interaction between various components of the TSSN. These results show that the TSSN architecture can be used

to monitor rail-borne cargo.

Portions of this chapter are the output of joint work done by the SensorNet team including: Daniel T. Fokum, Victor S. Frost, Martin Kuehnhausen, Daniel DePardo, Angela N. Oguna, Leon S. Searl, Edward Komp, Matthew Zeets, Daniel D. Deavours, Joseph B. Evans, and Gary J. Minden. Portions of this chapter were published in [49, 50, 53, 54].

3.1 Introduction

Chapter 2 presented work related to monitoring cargo in motion. This chapter studies field trials of a transportation security sensor network that monitors the integrity of cargo shipments in motion and reports any changes in the shipment integrity state to decision makers.

Work by the FBI [1] and Mayhew [2] has shown that cargo theft is a major problem. Most non-bulk cargo travels in shipping containers and container transport is characterized by complex interactions between shipping companies, industries, and liability regimes [3]. Deficiencies in the container transport chain expose the system to attacks such as the commandeering of a legitimate trading identity to ship an illegitimate or dangerous consignment, hijack, or the theft of goods. Insufficiencies in these areas can be overcome by creating secure trade lanes, or trusted corridors, especially at intermodal points, for example, at rail/truck transitions. Research and development is underway to realize the vision of trusted corridors.

The work described here focuses on: advanced communications, networking, and information technology applied to creating trusted corridors. The objective of the research is to provide the basis needed to improve the efficiency and security of trade lanes by combining real-time tracking and associated sensor information with ship-

ment information. One crucial research question that must be answered in order to attain this objective is how to create open technologies that will allow continuous monitoring of containers by integrating communications networks, sensors, as well as trade and logistics data. This integration must occur within an environment composed of multiple enterprises, owners, and infrastructure operators. Another objective was to gain experience with a rail environment and obtain realistic parameter values to support the modeling and system trade-off studies.

To achieve improved efficiency and security of trade lanes, we have developed a Transportation Security Sensor Network (TSSN) architecture, which uses Service Oriented Architecture (SOA) [55] principles, for monitoring the integrity of rail-borne cargo shipments. The TSSN is an open system where different components can be provided by different vendors. The TSSN is composed of a Trade Data Exchange (TDE) [4], Virtual Network Operations Center (VNOC), and Mobile Rail Network (MRN). The functions of each of these components are discussed in greater detail in Section 3.2. The TSSN detects events, integrates the event type from the train in the field with logistics information, and then reports those events that are important to decision makers using networks with commercial links. Decision makers want to be notified of events within 15 minutes [56] so that they can take effective action. For the TSSN to be deployed, we need to validate its architecture and understand the timeliness of the system response. Work by Arsanjani *et al.* [57] and Saiedian and Mulkey [58] shows that service oriented architectures introduce overhead. As a result, we want to determine whether an SOA-based system such as the TSSN provides timely event notification. TSSN event notification is also impacted by unpredictable packet latency on commercial networks and the use of e-mail and/or Short Message Service (SMS) [59] for event notification. Thus, we have designed and implemented

hardware and software needed to realize a prototype of the TSSN and carried out experiments [53] to characterize the system, particularly the end-to-end time between event occurrence and decision maker notification using SMS or e-mail as well as the impact of SOA overhead.

In this chapter we provide a high-level description of the TSSN architecture and document two field experiments that were carried out to demonstrate that sensors and software based on an open service-oriented architecture can be used to monitor cargo in motion. Our experiments focused on determining the time from event occurrence to decision maker notification as well as testing functionality between the component services of the prototype TSSN. Our experimental results show that decision makers can be notified of events on the train in a timely manner using the prototype TSSN architecture. The rest of this chapter is laid out as follows: In Section 3.2 we describe the TSSN architecture including the components and the prototype hardware implementation. Section 3.3 discusses experiments conducted to assess the suitability of the TSSN system for cargo monitoring. In Section 3.4 we discuss the framework used to post process the log files from our experiments. Section 3.5 presents empirical results showing the interaction between various components of the TSSN. In Section 3.6 we briefly discuss how the results from this chapter can be applied in models for analysis and design of communications systems for monitoring cargo in motion. Refinements to the TSSN given our field trial experiences are discussed in Section 3.7. Section 3.8 provides concluding remarks.

3.2 System Architecture

To achieve the vision of a trusted corridor we have designed and implemented a prototype of the Transportation Security Sensor Network (TSSN) architecture. The

detailed architecture of the TSSN, including system extensibility, is found in [52], whereas this section gives an overview of the TSSN. The architectural details discussed here are important in understanding the experiments and results presented in Sections 3.3 and 3.5, respectively.

The SOA and web services used in the TSSN enable the integration of different systems from multiple participating partners. Moreover, the use of SOA and web services enable data to be entered once and used many times. Using commercial off-the-shelf (COTS) hardware and networks, as well as an open systems approach, the TSSN is able to detect events and report those relevant to shippers and other decision makers as alarms. Furthermore, the TSSN supports multiple methods for notifying decision makers of system events.

The TSSN uses web service specification standards—such as Web Services Description Language (WSDL 2.0) [60], Simple Object Access Protocol (SOAP 1.2) [61], WS-Addressing [62], WS-Security [63], and WS-Eventing [64]—which are implemented through Apache Axis2 [65] and associated modules. These standards are used to exchange structured information between a web service and client. The use of SOAP allows the deployment of platform-independent interfaces and thus a heterogeneous network of web service platforms. On the other hand, since SOAP and web services are based on XML, which is verbose, there is communication and processing overhead related to SOAP messages.

The TSSN supports wireless and satellite communication technologies such as HSDPA (High-Speed Downlink Packet Access) [66] and Iridium [67]. The TSSN uses the Hypertext Transfer Protocol (HTTP) for message transport over wired and wireless links. Finally, the TSSN prototype uses sensors and readers from Hi-G-Tek [68]. There is also a need to gather log files to enable system debugging as well

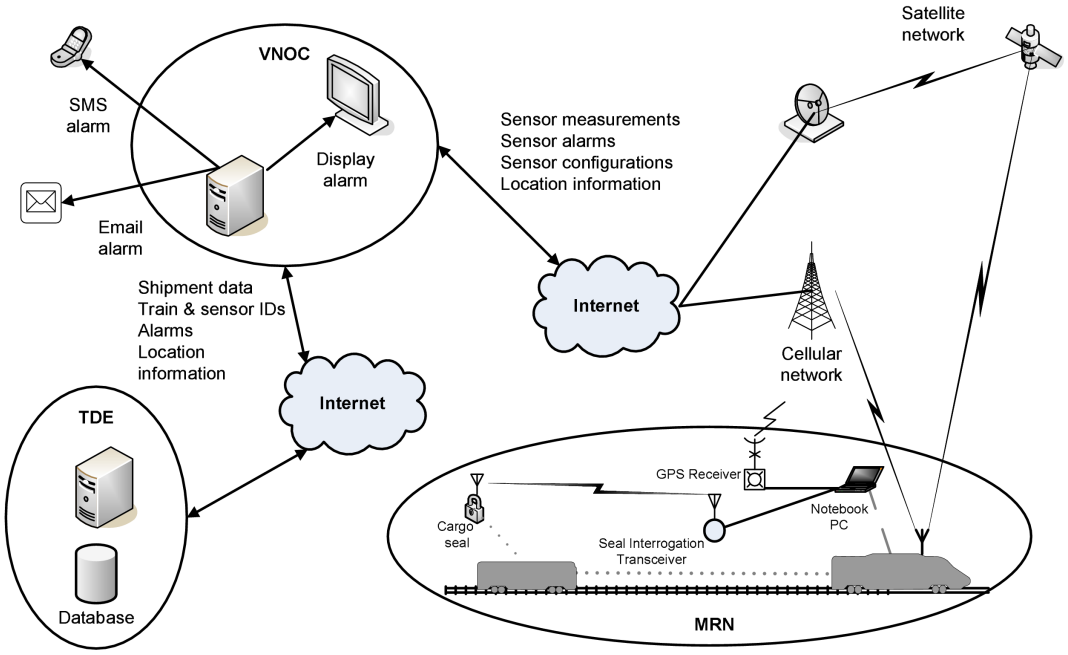


Figure 3.1. Transportation Security Sensor Network (TSSN) Architecture

as to capture metrics that can be used to evaluate system performance. Logging is currently done at the MRN, VNOC, and TDE using Apache log4j [69].

The TSSN system is composed of three major geographically distributed components: the Trade Data Exchange (TDE), Virtual Network Operations Center (VNOC), and Mobile Rail Network (MRN), as shown in Fig. 3.1. Wired links are used between the TDE and the VNOC, while MRN to VNOC communications are done using networks with commercial wireless link components. The TDE, VNOC, and MRN are examined in greater detail in the following subsections.

3.2.1 Trade Data Exchange

The Trade Data Exchange (TDE) contains shipping data and it interconnects commercial, regulatory and security stakeholders. The TDE is based on a "technology-neutral, standards-based, service-oriented architecture" [4]. The TDE is hosted on a

server with a wired connection to the Internet. The TDE is geographically separated from the VNOC and responds to queries from the VNOC. The TDE also stores event messages sent by the VNOC. Finally, the TDE sends commands to start and stop monitoring at the MRN as well as to get the train's current location.

In addition to the functions mentioned above, the TDE monitors the progress of shipment and other logistics information. The TDE captures commercial and clearance data including: the shipping list, bill of lading, commercial invoice, Certificate of Origin (for example, NAFTA Letter), and shipper's export declaration. It also validates and verifies data to ensure accuracy, consistency, and completeness. The TDE monitors the progress of the documentation and notifies responsible parties when errors or incompleteness pose the threat of delaying a shipment. The TDE forwards notification to the customs broker to request verification of the trade origination documents. The customs broker accesses the TDE via the same portal to review and verify the trade documentation. Finally, the TDE allows for collaboration between participating shippers, third-party logistics providers, carriers and customs brokers to define and document business requirements and risk assessment requirements. Real-time cargo sensing capability is provided to the TDE via the TSSN. Data from the TDE is combined with event data from the MRN to provide the decision maker complete information concerning the alarm, e.g., cargo information, location, and nature of the event.

3.2.2 Virtual Network Operations Center

The Virtual Network Operations Center is the management facility of the TSSN [52] and it is also the shipper's interface to the TDE. The VNOC can be the central decision and connection point for multiple MRNs. The VNOC consists of services

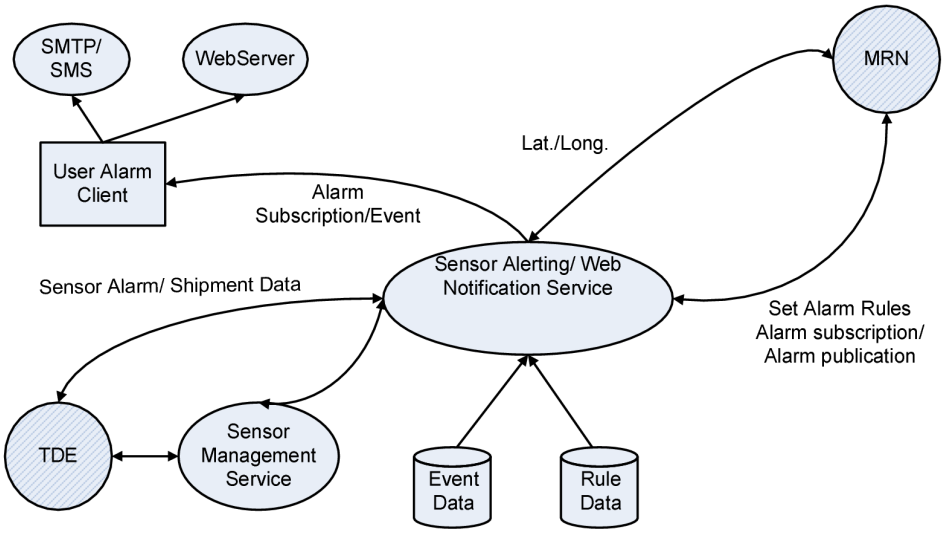


Figure 3.2. Virtual Network Operations Center Architecture

that receive and process alarms from the MRN as well as services that notify decision makers of events. Fig. 3.2 summarizes the VNOC and its components.

The functions of the VNOC include: forwarding commands from a client to the MRN to start and stop sensor monitoring as well as to get the MRN's current location, receiving *MRN_Alarms* from the MRN, obtaining event-associated cargo information from the Trade Data Exchange (TDE) in real time, and combining cargo information obtained from the TDE with an *MRN_Alarm* to form a *VNOC_Alarm* message that is sent (by SMS and/or e-mail to decision makers as shown in Figs. 3.9 and 3.10. A key role of the VNOC is getting the right alarm information to the right personnel in a timely manner and also to prevent personnel from being overwhelmed by event messages. An AlarmProcessor service in the VNOC makes decisions, using rules, on which *MRN_Alarms* are forwarded to decision makers. For example, a low battery alarm is sent to technical staff, while an unexpected open/close event is sent to system security personnel. These decisions are made using a complex event processor, Esper [70], which takes into account shipping information as well as data (for example,

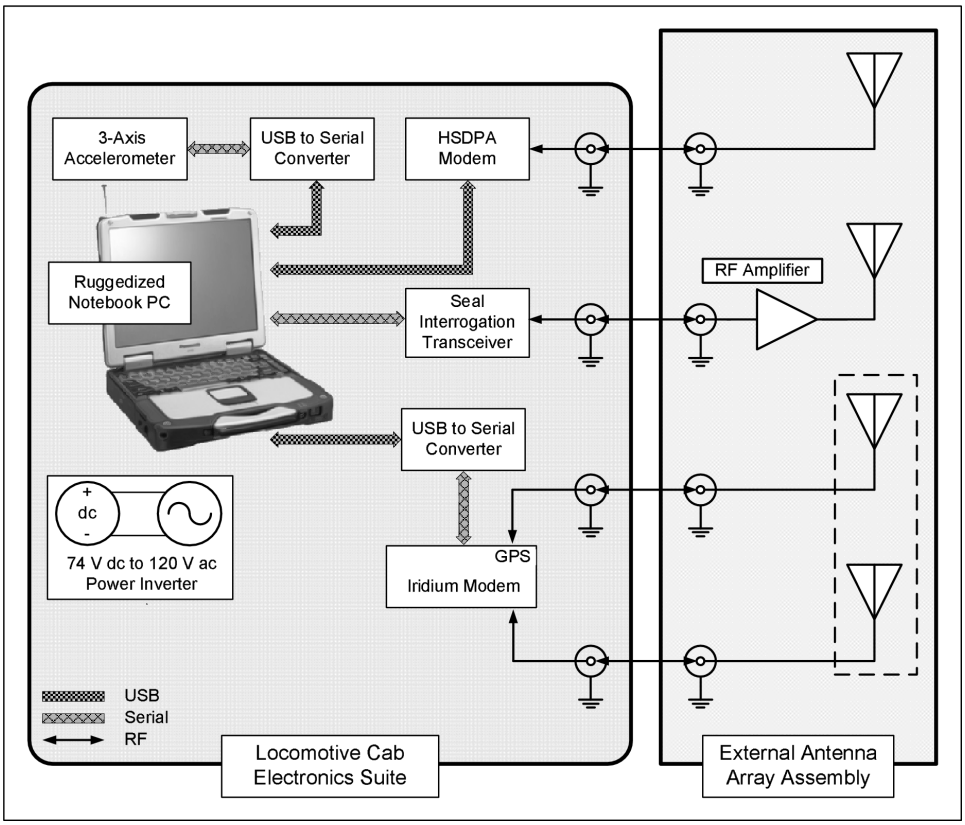


Figure 3.3. TSSN Collector Node Hardware Configuration

geographical location) from current and past *MRN_Alarms*.

3.2.3 Mobile Rail Network

The MRN subsystem is located on the train and it consists of hardware and software. The prototype hardware and software architecture is described below.

3.2.3.1 Mobile Rail Network Hardware

The MRN subsystem hardware consists of a set of wireless shipping container security seals and a TSSN collector node. The collector node is composed of two major sections: an electronics suite mounted in the locomotive cab and a remote antenna assembly that is magnetically attached to the exterior of the locomotive.

Fig. 3.3 summarizes the key components of the TSSN collector node.

The electronics suite contains a power inverter, a security seal interrogation transceiver, a computing platform, wireless data modems, a three-axis accelerometer, and a GPS receiver. The antenna assembly consists of three communications antennas, a GPS receiver antenna, and a bidirectional RF amplifier. Coaxial cables connect electronics suite devices to corresponding antennas.

Container physical security is monitored using a system that was originally designed for tanker truck security [68]. Container security is monitored with active and battery-powered container seals (sensors) equipped with flexible wire lanyards that are threaded through container keeper bar lock hasps as shown in Fig. 3.4. These seals had no support for multihop communications. The TSSN is designed to monitor and report security seal events including seal opened, seal closed, tampered seal, seal armed, seal missing, and low battery warnings. The seal interrogation transceiver (SIT) communicates with the container seals over a wireless network while the interrogation transceiver communicates with a notebook computer via a serial data connection. Each container seal contains a clock that is periodically updated from the seal interrogation transceiver, while the time on the SIT is updated from the notebook computer. The mechanism for time synchronization of the seals is outside the scope of this chapter.

In order to conserve energy the container seals are asleep most of the time [71]. About every three seconds the seals listen for commands from the interrogation transceiver; however, the frequency at which the seals listen for commands is configurable. If the sensors are instructed to listen for commands more frequently then their battery lifetimes are reduced, whereas longer intervals between interrogations result in longer battery lifetimes [71].

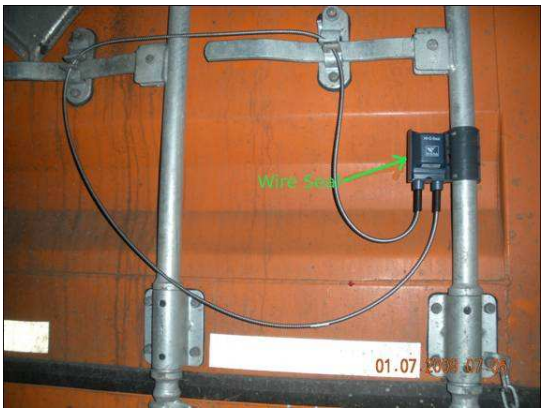


Figure 3.4. Container Seal

Communication between the MRN and the VNOC is accomplished using a HSDPA cellular data modem. An Iridium satellite modem is also available and is intended for use in remote locations that lack cellular network coverage. The Iridium modem is a combination unit that includes a GPS receiver, which is used to provide the MRN with position information.

3.2.3.2 Mobile Rail Network Software

The prototype MRN software was implemented using a service-oriented architecture approach. The software consists of a SensorNode service, an AlarmProcessor service, and a Communications service. The SensorNode service finds and monitors sensors that have been assigned to its control. The SensorNode service manages several sensor software plug-ins, for example, a seal interrogation transceiver plug-in and a GPS device plug-in, that do all the work on behalf of the SensorNode service. During typical operation each container seal listens for interrogation command signals at regular intervals from the interrogation transceiver. In the event of a seal being opened, closed, or tampered with, the seal immediately transmits a message to the SensorNode service running on the Collector Node. The message contains the seal

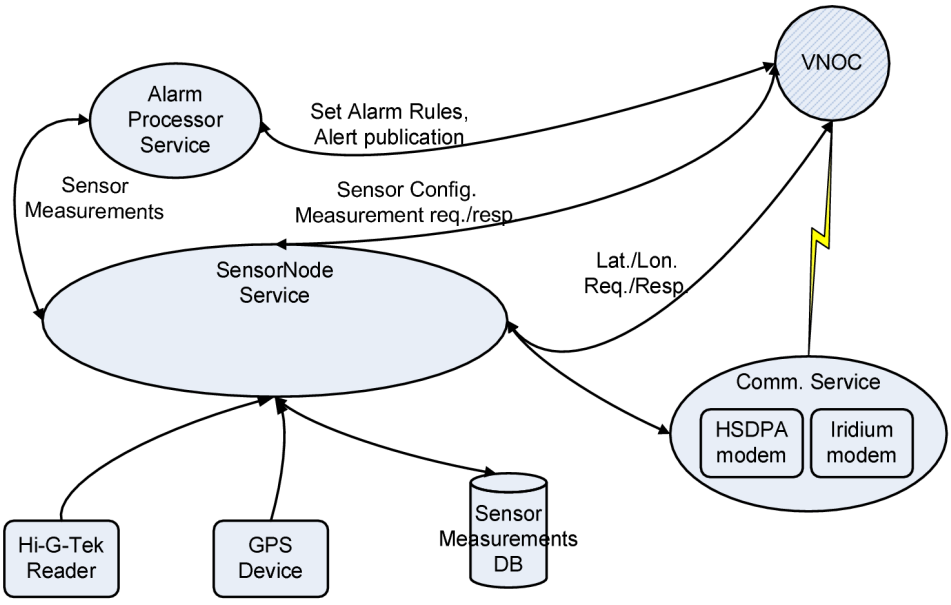


Figure 3.5. Mobile Rail Network Collector Node Architecture

event, a unique seal ID, and event time. The SensorNode service passes the seal message as an *Alert* message to the service that has subscribed for this information.

The AlarmProcessor service determines messages from the SensorNode service that require transmission to the VNOC. Alarm messages include the seal event, event time, seal ID, and train's GPS location. The Communications service uses either HSDPA or Iridium for reporting events via the Internet to the VNOC. Fig. 3.5 shows the key software functions of the MRN.

3.3 Experiments

This section presents two experiments—a road test and the short-haul rail trial—conducted to assess the suitability of the TSSN architecture for cargo monitoring as well as to collect data that would be used to guide the design of future cargo monitoring systems. It is non-trivial to carry out experiments on moving freight

trains; furthermore, as part of this effort we were limited to one chance to carry out experiments from a train. As a result, the TSSN architecture was tested in several static and some mobile tests, including the road test with trucks and the short-haul rail trial. In this section we present the experimental objectives, configuration, data collected during the tests, and issues that were encountered during the tests. The overarching goals of these experiments were to:

- Demonstrate the concept of using sensors, communications, and a service oriented architecture to monitor cargo in motion using the TSSN architecture.
- Determine the time from event occurrence to decision maker notification in a real field experiment.
- Verify proper operation of the prototype TSSN in a field environment. Proper operation means all messages were transmitted, received, and processed as expected and decision makers received all correct notification.

Thus, the following items were within scope of our experiments: the stability and timely performance of the communications protocols between TSSN component services, whereas the following items were out of scope: overall system robustness, whole train monitoring, energy consumption of the sensors, comprehensive security[1] issues, such as message spoofing, and decision maker response time given that event notification had been delivered.

3.3.1 Road Test with Trucks

The first experiment was conducted with two pickup trucks on local roads to validate the system operation and to determine if correct information is reported by

[1]Comprehensive security issues are being addressed in the next version of the prototype.

Figure 3.6. Map of Road Test with Event Annotations

the TSSN collector node, including valid GPS coordinates. One of the pickup trucks used in the test had the locomotive cab electronics suite in the truck bed, while both trucks had seals in their truck cabins so that seal open and close events could be emulated and reported. The VNOC was located in Lawrence, Kansas while the TDE was located in Overland Park, Kansas. The trucks were driven for approximately 1.5 hours over a 90 km route that began and ended in Lawrence, Kansas. The experiment route covered suburban and rural roads as well as state highways. During the experiment the seals were opened and closed at selected intersections along the test route that were easily identifiable on Google Maps [72]. Fig. 3.6 shows a trace of our route and the events overlaid on a Google map.

3.3.2 Short-haul Rail Trial

Another experiment was carried out using a freight train traveling from an intermodal facility to a rail yard. Our objectives in this experiment were the following:

- To determine the performance of the prototype TSSN architecture when detecting events on intermodal containers in a real rail environment.

- To investigate if decision makers could be informed of events in a timely manner

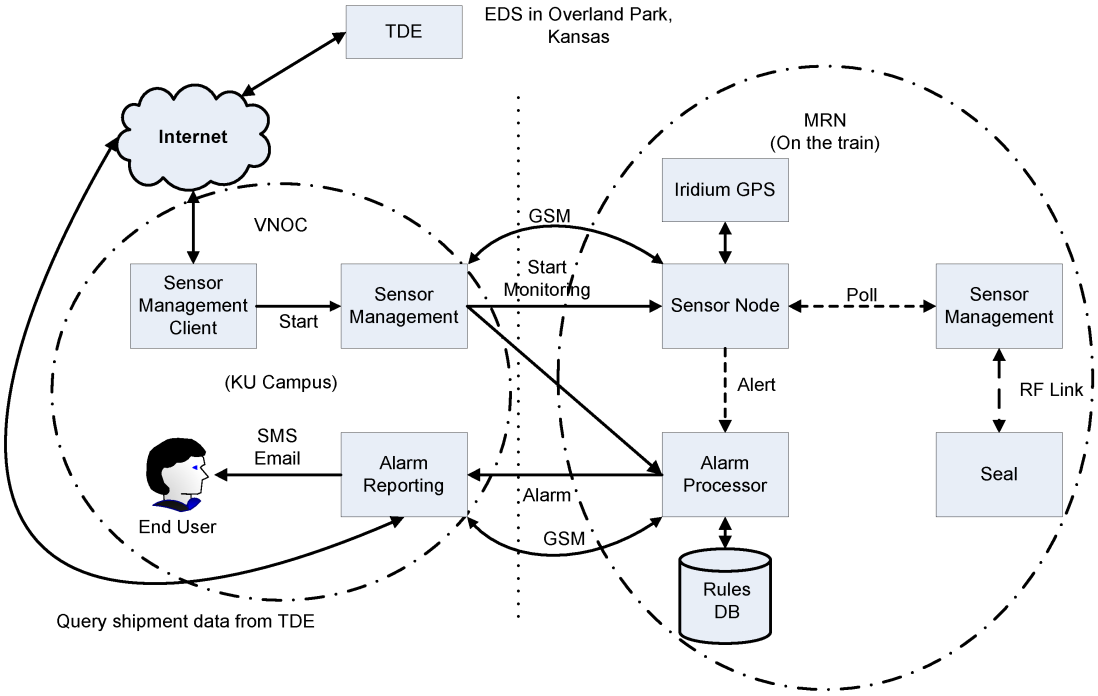


Figure 3.7. Logical Short-haul Rail Trial Configuration

using SMS messages and e-mails.

- To collect data that will be used in a model to investigate system trade-offs and the design of communications systems and networks for monitoring rail-borne cargo.

- Evaluate the overall system performance to guide the future development of the TSSN architecture.

Fig. 3.7 shows the logical system configuration used in the short-haul rail trial. In this experiment the VNOC was located in Lawrence, Kansas, the TDE was located in Overland Park, Kansas, and the MRN was placed on the train. Within the MRN, the TSSN collector node was placed in a locomotive and used to monitor five seals. All communications between the MRN and the VNOC were passed through a Virtual

Figure 3.8. Collector Node and Sensor Deployment during Short-haul Rail Trial

Private Network (VPN) for message security. Prior to the start of the experiment prototype logistics data was added to the TDE to facilitate testing.

During the short-haul trial the train traveled for approximately five hours over a 35 km (22 miles) route. The route, which traversed both rural and urban areas, was relatively flat with a total elevation change of about 100 m. Fig. 3.8 shows a picture of the train used in the short-haul rail trial along with the arrangement of the sensors (wire seals). As shown in Fig. 3.8, the short-haul trial train was composed of well-cars with a mixture of empty cars, cars with a single container, or cars with double-stacked containers. Since we were demonstrating a proof of concept and the sensors in use for this test were commercial-off-the-shelf devices with no support for multihop communications, three sensors were placed on containers on three of the five railcars nearest to the locomotive so that they could be within radio range of the seal interrogation transceiver. One sensor was placed on the front of the locomotive while the fifth sensor was kept in the locomotive and manually opened and closed while the train was in motion to create events.

During the experiment the VNOC reported events to decision makers using e-mail and SMS messages. The e-mail messages also include a link to Google Maps, so that

```
NOC_AlarmReportingService:
 Date-Time: 2009.01.07 07:12:17 CST /
    2009.01.07 13:12:17 UTC
 Lat/Lon: 38.83858/-94.56186,
          Quality: Good
 http://maps.google.com/maps?q=38.83858,-94.56186
 TrainId=ShrtHaull
 Severity: Security
 Type: SensorLimitReached
 Message: SensorType=Seal
           SensorID=IAHA01054190
           Event=Open Msg=
 NOC Host: laredo.ittc.ku.edu

Shipment Data:
 Car Pos: 3
 Equipment Id: EDS 10970
 BIC Code: ITTC054190
 STCC: 2643137
```

Figure 3.9. E-mail Message Received during Short-haul Trial

```
NOC_Alarm:
Time:2009.01.07 07:12:49 CST
GPS:38.83860/-94.56186
Trn:ShrtHaull
Sev:Security
Type:SensorLimitReached
Msg:Sensortype=Seal SensorID=IAHA01054190
    Event=Close
```

Figure 3.10. SMS Message Received During Short-haul Trial

the exact location of the incident could be visualized. Fig. 3.9 shows the content of one of the e-mail messages that was sent to the decision makers and Fig. 3.10 presents an example of an SMS message.

In Figs. 3.9 and 3.10, the sensor ID, latitude and longitude data, and event type come from the MRN, while the shipment data comes from the TDE. The VNOC combines these pieces of information into an e-mail message that also includes a link to Google Maps, so that the exact location of the incident can be visualized.

During the test the interrogation transceiver lost communication with the seals for a brief period along the route while the train was stationary and then regained communications once the train started moving. We believe that this loss of communications was due to electromagnetic interference. However, further investigations are needed to validate this claim.

The short-haul rail trial was a success as all seal events were detected and reported to decision makers using both e-mail and SMS messages. Extensive log files were collected during the test and they were post processed to obtain data on TSSN system performance. The results from post processing, which are reviewed in Section 3.5, show that the prototype system functioned as expected.

Following this experiment, analysis of event logs obtained from the MRN, VNOC, and TDE revealed that there was a significant amount of clock drift on the TSSN Collector Node during this relatively short trial. The time recorded at the VNOC for receipt of a message, in some cases, was earlier than the time recorded at the TSSN Collector Node for sending the message. Since time at the VNOC is controlled by a Network Time Protocol (NTP) [73] server, we conclude that the clock drift is occurring on the TSSN Collector Node. The clock drift problem was resolved in the next version of the TSSN by using a high performance GPS receiver to get high quality local time. Pulse per second (PPS) output from the GPS receiver was used as an input to the NTP server running on the TSSN collector node. It should be noted that in spite of the clock drift in the TSSN collector node we were able to correct for it in our data analysis by assuming that data from different parts of the TSSN is independent, e.g., the time taken to break a seal and generate an alert message is independent of the time taken to transfer a message from the MRN to the VNOC. As a result we can measure elapsed time in different epochs separately and characterize

performance of the TSSN prototype.

3.4 Post-Processing of Experimental Data

In this section we discuss the framework for post processing the results of our experiments. During the short-haul rail trial we recorded events in log files at the geographically distributed VNOC, MRN, and TDE. These log files contained data on message sizes, timestamps, event type, message type (incoming/outgoing) amongst other data elements. Our objective was to post process these files to evaluate the performance of the prototype TSSN.

Post processing of log files was accomplished using a Java library (LogParser) that was developed in-house. First, the library read in all available information in each log file including time, message size, from and to addresses, as well as the original SOAP message. Information from the MRN, VNOC, and TDE log files in this experiment was combined into a single collection of log entries. We expect that every message transmitted in the TSSN should result in at least two log entries—a transmit log entry (at the originating entity) and a received log entry (at the receiving entity). The LogParser library identified log entries as:

- Transmit-receive pairs, that is, the outgoing and incoming log entries with the same SOAP WS-Addressing [62].

- Couples, that is, SOAP request-response message pairs.

Fig. 3.11 shows the relationship between log entry couples and transmit-receive pairs. Suppose the TDE sends a message to the VNOC requesting the current MRN location. The circled "1" and "2" in Fig. 3.11 denote the log entries representing message transmission from the TDE and receipt of this same message at the VNOC.

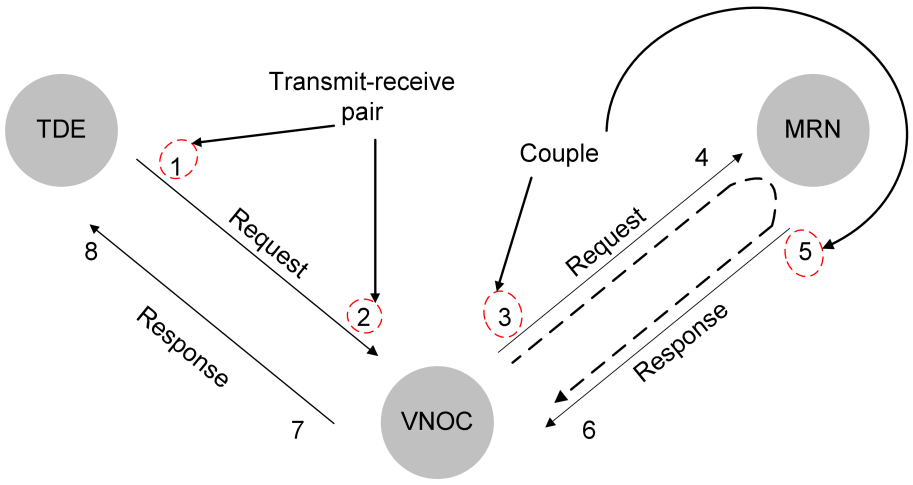


Figure 3.11. LogParser Framework Showing Message Couples and Transmit-receive Pairs

Much of the communication between client/server is based on a request-response model. As a result, there are two related messages which contain additional information to establish their relationship:

1. REQUEST: from client to server asking for something; and

2. RESPONSE: from server back to the client with the response.

Log entry couples are marked by the records for the outgoing request and response messages. Consequently, the circled "3" and "5" in Fig. 3.11 constitute the log entry couple for the VNOC forwarding the location request message to the MRN and the MRN's origination of a response respectively. Using the receive pairs for records "3" and "5", we can also identify entries "4" and "6."

With this framework, programs were written against the log entry collection to extract the number of messages sent by each service, request-response time for messages, processing time at either the MRN, VNOC, or TDE, the time that messages were carried by the network, and message sizes. Additional information, such as, latitude, longitude, sensor IDs, and event timestamps, is extracted from the SOAP

message using XPath expressions. XML Path language (XPath) is used for extracting information from XML by using path expressions that traverse the XML tree. Since SOAP is XML and the elements that we use, e.g., *Alerts*, *MRN_Alarms*, and *VNOC_Alarms*, are also XML, the use of XPath is appropriate. XPath also provides "basic facilities for manipulation of strings, numbers and booleans" [74].

3.5 Results

This section discusses the results of the experiments presented in Section 3.3. Most of the results shown here are based on the short-haul rail trial because we had more data to analyze. The results presented here are selected to test claims that:

- All messages between component services of the TSSN were transmitted, received, and processed as expected.
- Decision makers can be notified of events on the train in a timely manner.

The rest of this section is laid out as follows: Sections 3.5.1 and 3.5.2 present results on message counts for the road test and short-haul rail trial respectively. These results test the claim that all messages between component services of the TSSN are transmitted, received, and processed as expected. The rest of the results are based on the short-haul rail trial. Sections 3.5.3–3.5.5 study different portions of the time from event occurrence to decision maker notification to verify the claim that the TSSN can notify decision makers of events in a timely manner. Probability distributions are used in Section 3.5.6 to determine the likelihood of timely decision maker notification.

Note that due to significant clock drift in the TSSN collector node, we can only present an estimate of the time taken for an event report to travel from the MRN to

the VNOC. However, observed time values can be directly used for other TSSN component interactions. These results show how the aggregate time from event detection to decision maker notification is distributed among the various services and communication links in the TSSN. With this information we will be able to guide system refinements to further reduce the overall time. Suppose that T_n indicates when log entry n is made, then we can compute the following metrics:

- **Service request processing time.** This is the time between when a service receives a request and when a response message is composed. Using Fig. 3.11, this time is: $T_5 - T_4$.

- **Request-response time.** This is the time taken to get a response from a remote service, including the processing time. Using Fig. 3.11, this time is: $T_6 - T_3$.

- **Network time.** This is the time taken to get a response from a remote service, excluding the processing time. Using Fig. 3.11, this time is computed as: $T_6 - T_3 - (T_5 - T_4)$.

Our time analysis in Section 3.5.7 examines request-response messages from VNOC → MRN → VNOC, TDE → VNOC → TDE, and VNOC → TDE → VNOC.

The last objective of the short-haul rail trial was to collect data that will be used in a model [75] to support the future design of systems for monitoring rail-borne cargo and determine trade-offs. Message sizes are one component of this model. As a result, Section 3.5.8 presents a table summarizing the message size statistics between different components of the TSSN. It should be noted that message sizes can be computed *a priori*; however, the distribution of these messages cannot be determined beforehand.

3.5.1 Road Test: Message Counts

The primary goal of the road test was to validate TSSN prototype operation and to determine if correct information is reported by the TSSN collector node, including valid GPS coordinates. During the road test a manual record was made of all seal events and this written record was compared with the information generated from the TSSN. This comparison revealed that all open and close events were propagated correctly. During the approximately 1.5 hours long road test 76 messages (72 *Alarms*, 2 *StartMonitorSensors*, and 2 *StopMonitorSensors* commands) were exchanged on the VNOC to MRN link and these messages corresponded with the events that were recorded in the experiment log. Based on analysis of these messages we conclude that the system operated as expected. In addition, the experiment revealed that the TSSN was able to recover from a dropped HSDPA connection. However, it is worth noting that the seal interrogation transceiver was unable to read the sensors when the trucks were over 400 m apart on a hilly road. Based on the road test we conclude that the TSSN prototype worked as expected in a mobile scenario; we were able to combine sensor data from the MRN in a moving vehicle with shipment information obtained from the TDE to generate e-mail messages that were sent to distributed decision makers. Results from the road test showed that the TSSN prototype was ready for evaluation in a real rail environment.

3.5.2 Short-haul Trial: Message Counts

One objective of our post processing was to determine if messages were being passed correctly between the TSSN components. During the short-haul trial 203 messages (2 *StartMonitorSensors*, 2 *StopMonitorSensors*, 4 *SensorNodeStatus*, and 4 *SetMonitoringState* commands, 30 *getLocation* queries, 30 *Location* responses, and

131 *MRN_Alarms*.) were passed over the VNOC to MRN link. Full details on the messages exchanged are found in [49]. All of the *MRN_Alarms* received by the VNOC AlarmProcessor met the necessary rules so that they could be forwarded to decision makers as SMS and/or e-mail messages. The test users who were designated to receive all event notifications from the TSSN received 131 e-mail messages each.

3.5.3 Network Time from VNOC to MRN to VNOC

The network time statistics from VNOC to MRN to VNOC allow us to draw conclusions on the time taken to transfer request and response messages from the VNOC to the MRN and *vice versa*. These statistics also allow us to gain insight into the one-way network delay from the TSSN collector node on the train to the VNOC in Lawrence, Kansas—a delay that is one component of sending an *MRN_Alarm* message, which indicates an event at a sensor—from the MRN to the VNOC. Due to clock drift in the TSSN collector node, we are unable to obtain statistics on the one-way network delay from MRN $\rightarrow$ VNOC. However, it is reasonable to assume that the MRN $\leftrightarrow$ VNOC links are symmetric thus, the average one-way delay from the MRN to the VNOC is approximately 1.89 s. Fig. 3.12 is a histogram showing the network time for messages going from the VNOC to the MRN and back to the VNOC.

3.5.4 Elapsed Time from Alert Generation to AlarmReporting Service

The target notification time of security seal events is 15 minutes [56]. Thus, demonstrating that the elapsed time from alert generation to the AlarmReporting service is of the order of several seconds shows that the time taken to process events within the TSSN is not an impediment to timely notification. Fig. 3.13 shows the

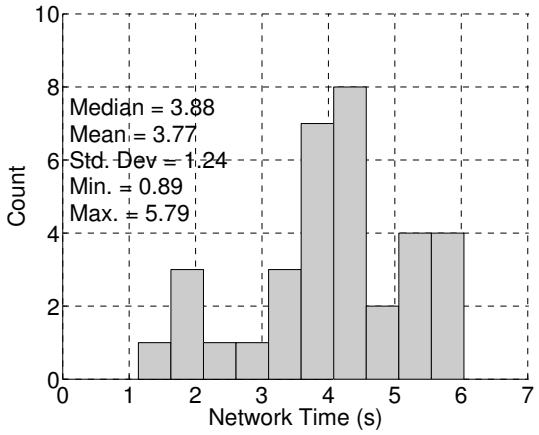


Figure 3.12. Network Times from VNOC → MRN → VNOC

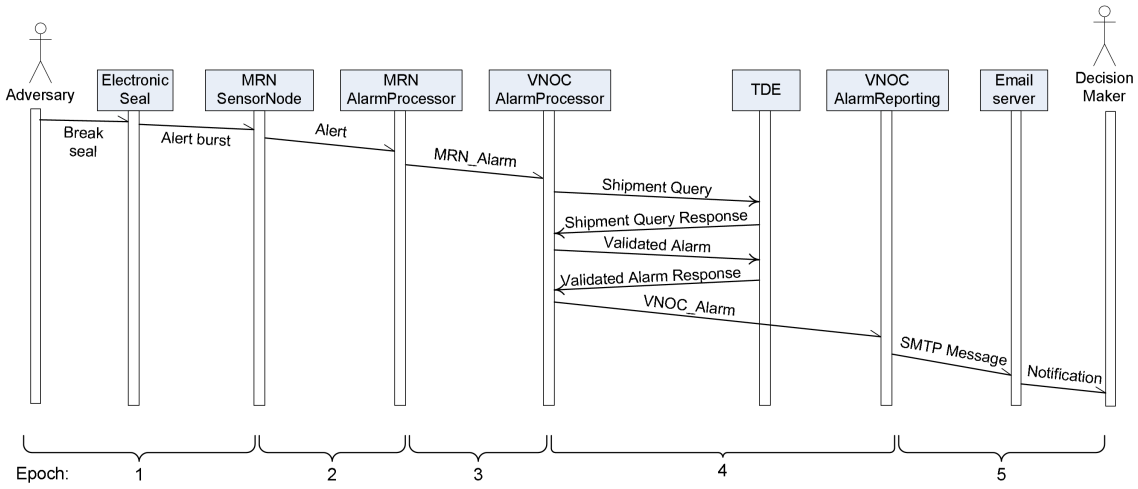


Figure 3.13. Sequence Diagram with Messages Involved in Decision Maker Notification

messages involved in notifying a decision maker of an event at a seal. This subsection deals with epochs 2, 3, and 4. Exact values can be computed for the time taken to propagate *Alert* and *VNOC_Alarm* messages, while we can use the 1.89 s estimate from the previous subsection as a reasonable value for the time taken to transfer a *MRN_Alarm* message from the MRN to the VNOC.

By analyzing the log files we see that on average it takes about 2 s for messages to get from the MRN SensorNode service to the VNOC AlarmReporting service. Thus,

Table 3.1. Summary of Time Statistics in Seconds for Decision Maker Notification

Epoch	Description	Min.	Max.	Mean	Median	Std. Dev.
1	Event occurrence to *Alert* generation	0.81	8.75	2.70	2.13	1.86
2	*Alert* generation to MRN AlarmProcessor service	0.01	0.08	0.02	0.01	0.01
3	One-way delay from MRN AlarmProcessor to VNOC AlarmProcessor	0.45	2.90	1.89	1.94	0.62
4	*MRN_Alarm* arrival at VNOC AlarmProcessor to AlarmReporting service	0.01	3.01	0.17	0.05	0.32
5	Elapsed time from VNOC AlarmReporting service to mobile phone	5.2	58.7	11.9	9.8	7.4

we conclude that the time taken to process events in the TSSN is not an impediment to timely notification of decision makers.

3.5.5 End-to-end Time from Event Occurrence to Decision Maker Notification

In this section we study the end-to-end system time between event occurrence and decision maker notification. The components of the end-to-end time include epochs 1–5 in Fig. 3.13. Decision makers are notified of events using SMS and/or e-mail. In the case of SMS notification a short SMTP message is sent to an e-mail-to-SMS gateway on a carrier's network, whereas with e-mail notification the SMTP message length is unrestricted and a message is sent to a e-mail server. The primary performance metric for prototype TSSN performance is the time between event occurrence until a decision maker is notified using an SMS message.

To gain an understanding of the end-to-end system time as well as overcome any clock errors in the MRN subsystem, we set up a laboratory experiment to determine the elapsed time between event occurrence and the TSSN's generation of the related event alert. In this experiment, a stopwatch was started when a seal was either broken or closed; when the MRN SensorNode service generated an *Alert* message the

stopwatch was stopped. From Table 3.1 we see that the longest observed time in epoch 1 is about 8.8 s, while the mean is about 2.7 s.

Since the commercial wireless networks used for decision maker notification are outside TSSN control, a second laboratory experiment was carried out to determine the elapsed time in epoch 5. In this experiment a client program was written to send messages to the VNOC AlarmReporting service. A stopwatch was started when the VNOC sent an alarm to a decision maker and the stopwatch was stopped when the decision maker's phone received an SMS message. This experiment was repeated for four different carriers resulting in the data shown in row 5 of Table 3.1.

From Table 3.1 we see that even though SMS was not designed as a real-time system, it provides excellent notification for this application, since most of our messages were delivered within one minute. Combining all of these results, we see that in these experiments the longest observed end-to-end system time was just over one minute[2] to notify decision makers of events. Most of this time is spent delivering an SMS message to the decision maker, so we conclude that the TSSN provides a mechanism for timely notification of decision makers.

3.5.6 Modeling of Decision Maker Notification Time

In this section we determine the likelihood of timely event notification. To determine the likelihood of timely event notification a probabilistic model is needed for the time epochs shown in Fig. 3.13. The observed histograms for each epoch visually resembled a Gamma distribution. Thus, in this analysis we assume the times in each epoch followed a Gamma probability density function. While the number

[2]This time is broken out as follows: in the longest observed times in our experiments it took approximately 8.8 s between event occurrence and the TSSN generating an alert; 2) it took approximately 4.91 s for an alert message to go through the TSSN until notification was sent to decision makers; and 3) it took up to 58.7 s to deliver an SMS message to decision makers.

Table 3.2. Estimated Gamma Distribution Parameters for Time Taken Between Seal Events and Decision Maker Notification

Epoch	Symbol	Description	$\hat{\alpha}$	$\hat{\theta}$
1	E_1	Event occurrence to *Alert* generation	4.01	0.60
2 + 4	$E_{2,4}$	*Alert* generation to MRN AlarmProcessor and *MRN_Alarm* arrival at VNOC AlarmProcessor to AlarmReporting service	1.13	0.13
3	E_3	One-way delay from MRN AlarmProcessor to VNOC AlarmProcessor	13.95	0.14
5	E_5	Elapsed time from VNOC AlarmReporting service to mobile phone	10.44	1.00

of observations (less than 130) was insufficient to statistically validate this assumption this postulate allows us to probabilistically determine if the TSSN prototype can provide event notification within 15 minutes [56] as required. The parameters for the distributions are estimated from the collected data and shown in Table 3.2, where $\hat{\alpha}$ and $\hat{\theta}$ represent the shape and scale parameters of the associated Gamma random variable. Let τ, which is composed of each of the epochs presented in Sections 3.5.3–3.5.5, represent the total time taken from event occurrence on the train to decision maker notification on a mobile phone. Nadarajah [76] states that the pdf of a sum of independent gamma random variables with parameters $(\hat{\alpha}_k, \hat{\beta}_k)$ is given by equation (3.1).

$$f_Z(z) = \frac{1}{\pi} \int_0^\infty \frac{\cos(\sum_{k=1}^N \arctan(\beta_k t) - zt)}{\prod_{k=1}^N (1 + t^2 \beta_k^2)^{\frac{\alpha_k}{2}}} dt \tag{3.1}$$

Then $\tau = E_1 + E_{2,4} + E_3 + E_5$ and we use the results from [76] to show that $\Pr[\tau \leq 240 \text{ sec}] = 99.9\%$. These results indicate that the prototype TSSN can notify decision makers in a timely manner with very high probability.

3.5.7 Timing Analysis of Other TSSN Interactions

Table 3.3 summarizes request/response, processing, and network time statistics for interaction between various TSSN components. The statistics on VNOC → MRN → VNOC interaction allow us to draw conclusions on request-response and processing times for certain (Start or stop monitoring at the MRN and get current MRN location.) VNOC commands. TDE → VNOC → TDE interaction statistics give us insight into the time taken to initiate and process commands to start or stop monitoring at the MRN, get the MRN's current location, or to process the setAlarmSecure command. The VNOC forwards these commands to the MRN and returns the MRN response to the TDE. To the TDE, all the elapsed time from when the VNOC receives a message from the TDE until the VNOC sends a response is processing time at the VNOC, even though part of that time is spent forwarding a request to the MRN and waiting for a response. Finally, the statistics on VNOC → TDE → VNOC interactions allow us to draw conclusions on request-response, processing, and network times for the TDE to store alarm messages and execute shipment queries. Both of these actions are carried out when the VNOC AlarmProcessor service is about to send an alarm to the VNOC AlarmReporting service. Note that there are no results for the MRN to VNOC to MRN interaction. This is for two reasons: first, clock drift in the MRN prevents us from computing a one-way network delay. Secondly, the MRN only generates response messages. As expected there are no request messages originating from the MRN that could be used in a log entry couple to calculate request-response or processing times.

Table 3.3. Summary of Time Statistics in Seconds for Other TSSN Interactions

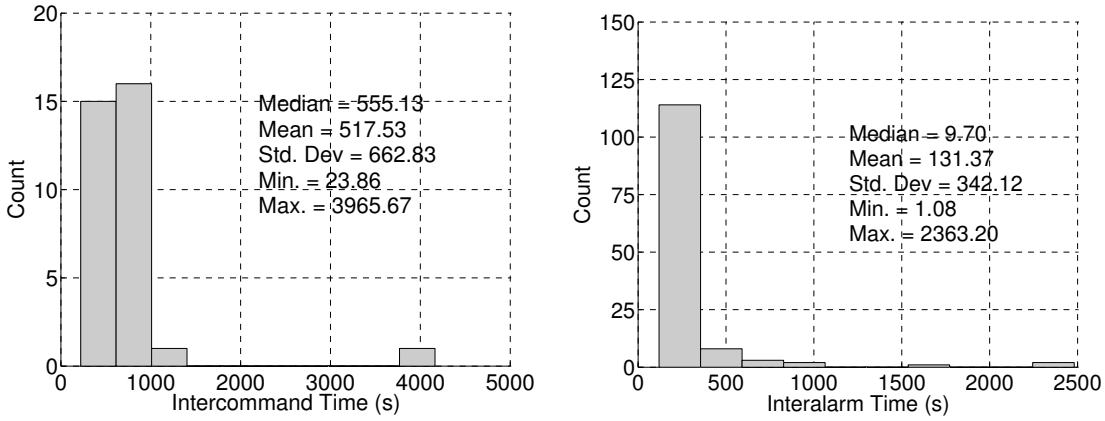
Description	Min.	Max.	Mean	Median	Std. Dev.
Request-response times from VNOC → MRN → VNOC	0.90	10.96	4.39	3.95	2.40
Network times from VNOC → MRN → VNOC	0.89	5.79	3.77	3.88	1.24
Processing times from VNOC → MRN → VNOC	0.00	5.21	0.61	0.01	1.69
Request-response times from TDE → VNOC → TDE	0.34	11.03	4.29	3.94	2.51
Network times from TDE → VNOC → TDE	0.00	4.00	0.14	0.04	0.64
Processing times from TDE → VNOC → TDE	0.29	10.98	4.15	3.85	2.45
Request-response times from VNOC → TDE → VNOC	0.02	0.41	0.12	0.07	0.11
Network times from VNOC → TDE → VNOC	0.01	0.08	0.05	0.07	0.02
Processing times from VNOC → TDE → VNOC	0.01	0.38	0.07	0.01	0.10

Table 3.4. Summary of Message Size Statistics in Bytes

Description	Min.	Max.	Mean	Median	Std. Dev.
TDE → VNOC	846	1278	874.7	848	96.8
VNOC → TDE	968	975	971.5	971	2.6
VNOC → MRN	650	1036	690.8	650	101.5
MRN → VNOC	799	1560	1419.2	1536	237.1

3.5.8 Message Sizes

Table 3.4 summarizes the message size statistics for all the messages exchanged in the TSSN. Message size data are needed for a model [75] that is under development to determine system trade-offs as well as optimal or near-optimal sensor locations when using a rail-borne cargo monitoring system. The cost of transmitting a message from the train to an operations center is one component of this model. This transmission cost, in turn, depends on the average message length transmitted from the train and the frequency at which these messages are generated.



(a) Intercommand (b) Interalarm

Figure 3.14. Intercommand and Interalarm Times at MRN

3.5.9 Intercommand and Interalarm Times

The data collected from these experiments will be used in a model to determine system trade-offs when using a rail-borne cargo monitoring system. Communication costs in this model depend on the frequency (interalarm time) with which messages need to be reported, the mode of communications, as well as the message length in bytes. The intercommand time is included in this analysis because incoming messages may also be billed. Figs. 3.14a and 3.14b summarize the intercommand and interalarm times respectively at the MRN. The data presented here can be used as a starting point for adaptive MRN Communications service algorithms that "call" the VNOC periodically.

3.6 Impact on System Modeling

New models are needed to characterize rail-based cargo monitoring systems. These models can be applied, along with optimization theory, to determine system trade-offs when monitoring cargo in motion. The models can also be used to find the best

locations for sensors in a rail-based sensor network as well as to guide the design of future cargo monitoring systems. In Section 3.5 we presented experimental results from a short-haul rail trial of the TSSN. In Chapter 4 we present models to determine optimal placement of sensors for monitoring rail-borne cargo. Our objective in this research is to develop extensible models that can give the best (cheapest) system design while preserving the shipper's desired level of security. Given a set, C, of containers to be placed on a train, a set, L, of possible locations for the containers on the train, a set, S, of sensors, and a set, R, of network elements, we can create a mapping, M_C, using Lai *et al.*'s [19] approach, that maps containers to locations on a train. We can also create mappings, M_R, and M_S, that map network elements and sensors, respectively, to locations on the train; alternatively, M_S may map sensors to containers. Given these mappings we can create a function, f, which takes as input the sets of containers, locations, sensors, and network elements, as well as the mappings described above and returns a system cost metric. The goal of this research is to develop such a function and use the results from Section 3.5 in making the model more realistic.

To this end two models have been built to compute the cost metric of a cargo monitoring system. The models have the following general format: Given a list of parameter values $p_1, p_2, \ldots, p_n$ (such as the container values, savings resulting from detecting events at containers, request/response times from VNOC $\rightarrow$ MRN $\rightarrow$ VNOC, and message sizes on the VNOC $\leftrightarrow$ MRN link), we define variables $x_1, x_2, \ldots, x_n$ (such as a variable that indicates if a sensor is placed on a certain container). We also define a function $f_o(\bar{x}; \bar{p})$ that depends on the parameters and variables to return the system cost. (One of the components of f_o includes the cost of transmitting event reports from the MRN to the VNOC.) Our goal in this research is to minimize

this objective function subject to the constraints[3] specified by the system designer. These models will be used to determine system trade-offs, such as a rail-mounted or trackside deployment of network elements.

3.7 Refinements Based on Experimental Results

This section proposes refinements to the TSSN based on experimental results. Recall from Section 3.3.2 that we have corrected the clock drift problem by using a high performance GPS receiver to get high quality local time on the TSSN collector node. In addition, post processing of the log files also indicated that a unique identifier—perhaps composed of a timestamp and counter—is needed in the *Alert*, *MRN_Alarm*, and *NOC_Alarm* messages to trace an *Alert* message through the TSSN. This identifier can also be used in the future to locate *MRN_Alarm* messages that need to be retransmitted to the VNOC following a loss of connectivity. Finally, the identifier can be used to mark previously processed messages so that the VNOC does not process the same message more than once.

Additional TSSN enhancements include:

- Redesigning the MRN hardware so that the TSSN collector node has redundant backhaul communication capabilities, for example, multiple satellite and cellular modems each with a different provider.

- Creating a comprehensive security framework for the TSSN. Ongoing research is addressing this issue [77].

- Enhancing sensor capabilities so that sensors can engage in multihop commu-

[3]Some of these constraints specify valid placements for sensors and associated communications infrastructure. The constraints might also require that events at certain containers be detected with a certain probability and reported within a given time interval with specified probability.

nications to enable whole-train monitoring.

The desired result of the research presented here is a standards-based open environment for cargo monitoring with low entry barriers to enable broader access by stakeholders while showing a path to commercialization.

3.8 Conclusion

In this chapter we have presented results from field trials of the prototype TSSN (Transportation Security Sensor Network). The TSSN is an open system where different vendors can supply different components of the system. Within the TSSN framework we have successfully combined sensor and shipment information to provide event notification to distributed decision makers. This chapter has shown results documenting the interactions between the different components of the TSSN. Based on our experiments and evaluations with the prototype the TSSN architecture is viable for monitoring rail-borne cargo. We have successfully demonstrated that alert messages can be sent from a moving train to the VNOC and combined with cargo information that is forwarded to geographically distributed decision makers using either SMS or e-mail. Furthermore, based on the experiments reported here, we are able to detect events and notify decision makers in just over one minute. Thus, we conclude that the TSSN architecture provides a mechanism for timely notification of decision makers. However, additional development and testing is needed before the TSSN architecture can be deployed in production systems. In Chapters 4 and 5 the results from this chapter will be used in models to determine optimal sensor placement on trains for cargo monitoring.

Chapter 4

Modeling for Analysis and Design of Communications Systems and Networks for Monitoring Cargo in Motion along Trusted Corridors

4.0 Chapter Summary

Exports from Asia to the United States have increased significantly in recent years, causing congestion at ports on the Pacific coast of the United States. To alleviate this congestion, some groups want to ship goods by rail directly from ports to inland intermodal traffic terminals. However, for such an effort to succeed, shippers must have "visibility" into the rail shipment. In this research we seek to provide visibility into shipments through optimal placement of sensors and network elements. We formally define visibility and then develop models to identify and locate network elements and containers on trains. Two models have been developed—one for use when all network elements are on the train and the other for use when some are located trackside—to determine sensor placements and network design. The models show that, under reasonable assumptions, sensor deployment reduces the overall system cost; therefore, sensor networks make sense for monitoring cargo. These models also enable the study of system trade-offs while achieving the desired level of visibility

into cargo shipments.

4.1 Introduction

As was argued in Chapter 1, cargo theft is a major problem affecting the shipping industry in the United States. We propose to alleviate this problem by providing visibility into cargo shipments through the placement of sensors on shipping containers. If the containers are tampered with, the sensors would send a message to an operations center using the Transportation Security Sensor Network described in Chapter 3. The objective of this chapter is to develop models that will allow for the best placement of sensors on containers while satisfying the shipper's visibility constraints. Furthermore, the models developed in this chapter will be used in Chapter 5 to study the system trade-offs that can be made while providing visibility into cargo shipments.

4.1.1 Visibility

In this section we provide a formal definition of visibility. Informally the integrity of a cargo shipment has state. These states will include locking the container and closing the seal, opening the seal and then the container, and tampering with the seal. Critical events are generated whenever the integrity of a shipment changes states. Examples of critical events include messages indicating that seals are opened, closed, or tampered with. The seals also generate other events during normal operations. These messages denote maintenance events and examples include alerts indicating an armed seal or a seal warning of a low battery. The set of messages will also include items such as a seal incorrectly reporting a tamper event, incorrectly being detected as missing, or an incorrect low battery report. This latter group of events will be

considered false alarms. Important aspects of visibility include the likelihood of a sensor detecting an event at a container, the time taken by a sensor to notify decision makers of an event, and the likelihood of a false alarm from a sensor. We define visibility as a binary variable that relates the probability of detecting an event at a container with the time taken to report that event to the decision maker and the probability of false alarm for that container. More formally we define a container j, as visible if $\nu(j,\, t,\, \tau,\, \mathrm{TR}_j,\, P_\epsilon,\, E_j,\, P_\alpha,\, F_j) = 1$, where the visibility function is defined as in equation (4.1) and the parameters of the function are:

- An event can be detected at container j, and made known to the decision maker with a probability P_ϵ, that is greater than or equal to some threshold, E_j.
- The time t, taken to notify the decision maker of an event, must lie within an interval of length τ, with probability greater than or equal to some threshold TR_j, i.e., $\Pr(t \leq \tau) \geq \mathrm{TR}_j$.
- The probability of false alarm at container j, P_α, must be kept less than or equal to some threshold F_j.

The system design determines P_ϵ, P_α, and $\Pr(t < \tau)$. The combination of P_ϵ, P_α and $\Pr(t < \tau)$ can be mapped into a visibility space.

$$\nu(j,\, t,\, \tau,\, \mathrm{TR}_j,\, P_\epsilon,\, E_j,\, P_\alpha,\, F_j) = \begin{cases} 1 & \text{if } (\Pr(t \leq \tau) \geq \mathrm{TR}_j \text{ AND} \\ & \quad P_\epsilon \geq E_j \text{ AND } P_\alpha \leq F_j) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

4.1.2 Metrics

Metrics are needed to compare the "goodness" of two or more proposed system designs. In this section we present our metrics, which include:

- **System operational cost.** This metric is computed per trip, and it consists of each sensor's false alarm cost, the cost of deploying the sensors, repeaters and readers, network, and the backhaul communications devices, as well as the cost of reporting events. The costs of missing an event at a given container as well as the costs of a communications failure at a sensor are also components of this metric.

- **Visibility metric.** We declare that container j is visible if the sensor placed on j meets all the system designer-imposed constraints for visibility of the container.

The rest of this chapter is laid out as follows: in Section 4.2 we present a scheme for identifying containers, sensors, and the locations that they occupy on trains. Section 4.3 discusses two possible deployments for cargo monitoring systems. In Section 4.4 the parameters and variables in the models for analysis and design of communications systems for cargo monitoring are introduced. The models for optimal sensor and communications system assignment are described in Section 4.5. Section 4.6 presents arguments for validating the models as well as discussing model growth. Concluding remarks are provided in Section 4.7.

4.2 A System Description for Identifying and Locating System Elements

Notation is needed to identify sensors and containers in models to analyze and design communications systems and networks for monitoring cargo in motion along

trusted corridors. In this section we present one scheme for identifying containers and the locations that they occupy on a train. We focus first on container identification and then turn our attention to indexing container and sensor locations on trains.

4.2.1 Identification

The containers that are to be placed on the train typically come in a variety of lengths ranging between 20 ft. (6.1 m) and 53 ft. (16.2 m). We propose sorting the loads by length and numbering each container with a unique index j, that is sufficient for accessing the container's properties, such as its length, weight, value, and other intrinsic attributes of the container that might be necessary to solve the sensor placement problem and identify system trade-offs. For example, suppose we have two 20 ft. (6.1 m) containers, two 40 ft. (12.2 m) containers, and one 45 ft. (13.7 m) container, then j ranges from $1 \ldots 5$, with each container bearing a unique j index. The container types can then be identified by using a function that returns the length of a container when given an integer j, i.e., the container lengths could be stored in a vector L, so that L_j indicates the length of container j.

Every sensor that is to be placed on the containers is identified with a unique index, i. This index, which starts off with value 1, is sufficient for reading the parameters, e.g., transmission range, associated with each communications element.

4.2.2 Location

Each railcar consists of one or more permanently attached units, where a unit is a frame that can support one or more slots [19]. Each unit is uniquely identified by an integer k, where k starts off with value 1. The index $k = 0$ is reserved for the

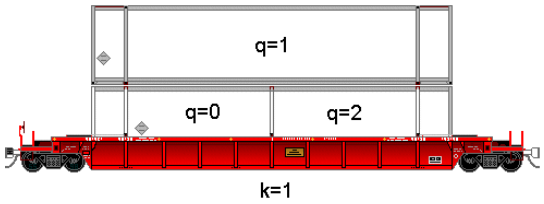


Figure 4.1. Unit with Two 20 ft. Containers and One 40 ft. Container

locomotive.[1]

Review of the Association of American Railroads Loading Capabilities Guide [78] indicates that railcars used for intermodal transportation have at most two positions (layers) for carrying intermodal loads. Within each position, slots are available for holding containers. For example, [78] indicates that two 20 ft. containers can be placed in the bottom position and a 40 ft. container is placed over both 20 ft. containers in the top position, as shown in Fig. 4.1.

In general, the first available slot in a unit, i.e., the first slot in the bottom position is marked with index $q = 0$. All other slots in the bottom position have even q indices. The first slot in the top position is always indexed with index $q = 1$, while all other slots in the top position will have odd indices.[2]

From above we see that the integer triple (j, q, k) is sufficient for identifying the unit and slot occupied by container j. For instance, using the five container example presented in Section 4.2.1, the integer triple (1, 0, 2) implies that container 1 is found in slot 0 of unit 2. Similarly, (5, 1, 1) implies that container 5 is found in slot 1 of unit 1.

In this section we presented an orthogonal indexing system for containers on a

[1]The issue of having to deal with several locomotives on a train is not part of this model. If more than one locomotive is present, all the locomotives are treated as one with respect to the goals of this system.

[2]A review of the Association of American Railways Loading Guide [78] indicates that we will rarely have more than one container in the top position.

train. This numbering scheme is based on assigning containers with a unique integer j, which is used to identify containers and to retrieve additional container attributes, an index k, which is used to identify units, and an integer q, used to identify slots on a unit.

4.3 System Deployment Scenarios

This section discusses the two deployment scenarios for cargo monitoring systems and presents some of the high level issues associated with each system. For each system deployment scenario it will be assumed that the containers have already been placed into slots on the train. For the train-mounted system deployment case there will be one backhaul communications device on the train. The objective in this case will be to assign sensors to the containers such that the visibility constraints are satisfied and the system cost metric is minimized. For the trackside system deployment case there will not be any backhaul communications device on the train. Instead, there will be readers with backhaul communications capability at regularly spaced intervals along the track. The trackside reader separation interval will be subject to the deadline for event notification and train speed. As was the case for the train-mounted model, the objective is to assign sensors to the containers such that the visibility constraints are satisfied.

The issues raised within each system deployment case are outside the scope of this paper, however, they are mentioned very briefly below:

- For the train-mounted system deployment all of the system elements are stationary with respect to each other, whereas with the trackside system the sensors are moving past the readers. As a result, this adds a level of complexity to the trackside system.

- Start-up costs for the trackside system will be much higher than those for the train-mounted system. This is because only one backhaul communications device is needed for the train-mounted system, whereas several readers, including backhaul communications devices, are required for the trackside system deployment.

Section 4.4 discusses the parameters and variables associated with each of the system deployment scenarios.

4.4 Parameters and Variables

In this section we use the container identification and location scheme from Section 4.2 to introduce the parameters and variables in two models for computing the cost metric for cargo monitoring systems. Parameters will be given to the system designer for a specific placement problem, while the optimization process will assign appropriate values to the variables such that the objective is minimized while satisfying any design constraints. To facilitate the presentation of the variables and parameters, throughout this section we use the five container example shown in Fig. 4.2. The rest of this section is laid out as follows: Section 4.4.1 introduces the parameters for the models. For the sake of completeness there is a very brief discussion on container assignment parameters in Section 4.4.1.1 while the communications system parameters are presented in Section 4.4.1.2. Section 4.4.1.3 uses probability distributions to determine the likelihood of timely decision maker notification. The variables for the models are introduced in Section 4.4.2.

Suppose that we have a train with a locomotive and two well cars[3] as shown in Figure 4.2. Furthermore, assume that we have two 20 ft. containers, two 40 ft.

[3]A well car is a type of railcar with a depressed section between the wheels for holding intermodal containers.

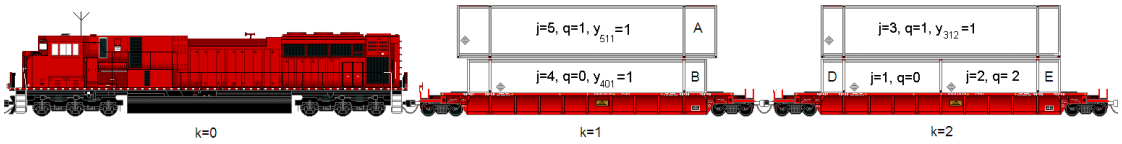


Figure 4.2. Two Well-cars with Load Indices Identified

containers, and one 45 ft. container. Recall that each container (load) is uniquely identified by an integer j, while the units are uniquely identified by an index k. Assume that the containers are indexed[4] such that $j = 1$ refers to the most expensive 20 ft. container, while $j = 2$ refers to the least expensive 20 ft. container, $j = 3$ and $j = 4$ refer to the 40 ft. containers. Finally, $j = 5$ is used to denote the 45 ft. container.

4.4.1 Parameters

This section introduces the parameters for models. First, we discuss the container assignment parameters, which indicate valid container assignments as well as information on container and unit attributes. Next, we discuss the communications systems assignment parameters. Finally, we present some distributions to model the time taken to notify decision makers of events on a train.

4.4.1.1 Container Assignment Parameters

This section introduces the container assignment parameters for our model. The length of the k^{th} unit is represented by U_k and the length of the j^{th} container is given by L_j. The binary parameter y_{jqk} indicates a given container's location on the train,

[4]Note that we are not required to index the containers by value. The indices could be randomly assigned as long as each number is used exactly once.

and it is defined as:

$$y_{jqk} = \begin{cases} 1 & \text{if } j^{\text{th}} \text{ container is assigned to slot } q \text{ in unit } k, \\ 0 & \text{otherwise} \end{cases}$$

4.4.1.2 Communications Systems Assignment Parameters

In this section we introduce parameters that are necessary to address the system design, including the sensor assignment portion of the container assignment and sensor assignment problem.

Suppose each of the containers has a value v_j, furthermore, suppose that each time the decision maker receives notification of an event at a container within a time interval τ_j we get a savings σ_j (Note that σ_j can be greater than v_j.). These savings can be viewed as the value of a detected event to the decision maker. We assume that all the repeaters and backhaul communications devices on the train are arranged in a linear topology.

Backhaul communications devices in our system are used to transmit event reports from the train to the decision maker (possibly via an operations center). We use the binary parameter B_{qk} to indicate when a backhaul communications device is placed in slot q of unit k. This parameter is defined as:

$$B_{qk} = \begin{cases} 1 & \text{if backhaul communications device is placed} \\ & \text{in slot } q \text{ of unit } k, \\ 0 & \text{otherwise} \end{cases}$$

In our system, sensors have a limited transmission range, and they are interrogated by more powerful radios called "repeaters/readers." The repeaters can communicate

Table 4.1. Train-related Parameters

Parameter	Comment
D	Rail trip duration in hours.
d_T	Length of rail journey in kilometers.
$\dot{x}$	Train-speed in kilometers per hour.
t_f	Number of trains passing a given trackside reader per hour.
t_L	Number of trips per locomotive per hour.
ζ	Probability of event occurrence during a trip.
Z	Probability of the train being stationary.

Table 4.2. Sensor and Communications Equipment-related Parameters

Parameter	Comment
LT_A	Useful lifetime of trackside reader in hours.
LT_c	Useful lifetime of cellular communications device in hours.
LT_s	Useful lifetime of satellite communications device in hours.
FP_2	Weight of sensor cost allocated to improving event detection.
FP_3	Weight of sensor cost allocated to improving timely reporting or successful communications in train-mounted and trackside cases, respectively.
FP_4	Weight of sensor cost allocated to reducing false alarms.
FP_5	Weight of sensor cost allocated to improving sensor transmission range.
FP_6	Weight of sensor cost allocated to reducing sensor read time.

with each other over longer distances to get event reports to a backhaul communications device. We use the binary parameter A_{qk} to indicate when a repeater is placed in slot q of unit k. This parameter is defined as:

$$A_{qk} = \begin{cases} 1 & \text{if repeater is placed in slot } q \text{ of unit } k, \\ 0 & \text{otherwise} \end{cases}$$

There are other communications systems assignment parameters used in the optimal placement model. These parameters are shown as follows: Table 4.1 presents train-related parameters, sensor and communications equipment-related parameters are listed in Table 4.2, Table 4.3 presents message-related parameters, communications system probability parameters are defined in Table 4.4, and all the cost parameters in the model are listed in Table 4.5.

Table 4.3. Message-related Parameters

Parameter	Comment
l	Average message length in bytes between sensor and operations center.
λ_i	Message generation rate for sensor i.
RTT_c	Communications round trip time in seconds from train to operations center over the cellular link.
RTT_s	Communications round trip time in seconds from train to operations center over the satellite link.

Table 4.4. Communications System Probability Parameters

Parameter	Comment
$\Pr(H)$	Probability of successfully transmitting a message from the train to the operations center over the cellular link.
$\Pr(I)$	Probability of successfully transmitting a message from the train to the operations center over the satellite link.

Table 4.5. Cost Parameters

Parameter	Comment
C_α	Cost of one false alarm.
C_s	Cost of sending one byte by satellite.
C_c	Cost of sending one byte by cellular.
C_A	Acquisition cost of one reader/repeater.
C_F	Fixed cost of acquiring a sensor.
C_{BC}	Acquisition cost of one backhaul communications device (cellular).
C_{BS}	Acquisition cost of one backhaul communications device (satellite).
C_{HL}	Installation cost of one sensor/seal.
C_{AL}	Installation cost of one reader/repeater.
C_{AD}	Installation cost of one trackside reader.
C_{BD}	Installation cost of one trackside cellular communications device.

4.4.1.3 Distributions for Decision Maker Notification

The results from Section 3.5.6 will be used to model the time taken for decision maker notification. These results will be used to determine values for $\Pr(t < \tau)$, the probability that a sensor notifies a decision maker in a timely manner.

4.4.2 Communications Systems Assignment Variables

This section presents the variables used to indicate communications system assignment in our models. These variables are either integers or positive real numbers. Appropriate values will be assigned to these variables such that the best objective function value is attained. First, we present the variable that is common to the track-side and train-mounted cases. Next, we present the other variables that are unique to each case. In general, whenever a variable or parameter is indexed by $q = 0$, and $k = 0$ it is assumed that we will be referring to the locomotive. For example, $A_{00} = 1$ and $B_{00} = 1$ will indicate that a reader and a backhaul communications device, respectively, are located on the locomotive. In our discussion, a "sensor" refers to the combination of sensing and communication devices, e.g., the seal shown in Fig. 3.4. The binary variable S_{ijqk} indicates when sensor i is assigned to the j^{th} container. This variable is defined as:

$$S_{ijqk} = \begin{cases} 1 & \text{if sensor } i \text{ is attached to } j^{\text{th}} \text{ container in slot } q \text{ of unit } k, \\ 0 & \text{otherwise} \end{cases}$$

4.4.2.1 Train-Mounted Deployment Variables

There are other variables, in addition to S_{ijqk}, used for the case when the sensors and related communications infrastructure are on the train. Table 4.6 presents these variables and equations (4.2)–(4.9) show how the variables are computed.

$$\Gamma_k = C_\alpha \sum_{\forall\, i,j,q} \alpha S_{ijqk} y_{jqk} \tag{4.2}$$

Table 4.6. Train-Mounted Deployment Variables

Variable	Comment
α	Probability of false alarm for sensor.
ϵ	Probability of event detection by sensor.
φ	Probability of successful end-to-end communications from sensor to operations center.
C_H	Acquisition cost of one sensor/seal.
Γ_k	Cost of false alarms per unit.
Δ_k	Cost of missed detection per unit.
Ξ_k	Cost of reporting an event outside of desired deadline for container visibility.
Λ_k	Cost of transmitting messages generated by all the sensors on a unit.
Ψ_k	Cost of acquiring and installing sensors on each unit.
Υ_k	Cost of acquiring and installing repeaters on each unit.
Ω_k	Cost of acquiring and installing one backhaul communications device on each unit.

$$\Delta_k = \zeta\left(\sum_{\forall j,q} \sigma_j y_{jqk} - \sum_{\forall i,j,q} \epsilon\sigma_j S_{ijqk} y_{jqk}\right) \tag{4.3}$$

$$\Xi_k = \zeta\left(\sum_{\forall j,q} \sigma_j y_{jqk} - \sum_{\forall i,j,q} \varphi\sigma_j S_{ijqk} y_{jqk}\right) \tag{4.4}$$

$$\Lambda_k = D(\Pr(H)C_c + \Pr(I)(1-\Pr(H))C_s)l \sum_{\forall i,j,q} \lambda_i S_{ijqk} y_{jqk} \tag{4.5}$$

$$C_H = C_F + \epsilon \mathrm{FP}_2 + \varphi \mathrm{FP}_3 + (1-\alpha)\mathrm{FP}_4 \tag{4.6}$$

$$\Psi_k = \sum_{\forall i,j,q} (C_H + C_{HL}) S_{ijqk} y_{jqk} \tag{4.7}$$

$$\Upsilon_k = \sum_{\forall q} (C_A + C_{\mathrm{AL}}) A_{qk} \tag{4.8}$$

$$\Omega_k = \left(\frac{C_{\mathrm{BC}}}{t_L \times \mathrm{LT}_c} + \frac{C_{\mathrm{BS}}}{t_L \times \mathrm{LT}_s}\right) \sum_{\forall q} B_{qk} \tag{4.9}$$

The cost of false alarms per rail car is given by equation (4.2). This is given by the cost of each false alarm times the sum of probabilities of false alarm for all the sensors that are currently used. Assume that if an event is detected and reported in a timely manner, then there is no loss to the decision maker. In addition, assume that

the probability of an event occurring at a container is independent of the probability of a sensor detecting that event or reporting it in a timely manner. Equation (4.3) computes the cost of a missed detection per unit, which is given by the probability of event occurrence times the savings that are lost if an event is not detected. Similarly, equation (4.4) computes the cost of reporting an event outside the required deadline for container visibility. This cost is given by the probability of event occurrence times the savings that are lost if the event is not reported in a timely manner. Equation (4.5) computes the cost of transmitting messages generated by all the sensors on a unit. This cost is given by the rail trip duration times the mean cost of transmitting one byte times the sum of message generation rates for all sensors in use. The unit cost of acquiring a sensor for the train-mounted deployment is captured in equation (4.6). This cost is given by adding up the fixed cost of acquiring each sensor, plus the cost of getting a sensor with specified probabilities of detection, timely reporting, and false alarm. The cost of acquiring and installing the sensors on a unit is given by substituting equation (4.6) into (4.7). Repeater acquisition and installation costs per unit are computed with equation (4.8). Finally, equation (4.9) calculates the cost of acquiring and installing a backhaul device on each rail car. We assume that the backhaul devices are reused for several trips, thus we amortize this cost over the expected number of trips in the device's lifetime.

4.4.2.2 Trackside Deployment Variables

The variable S_{ijqk}, which is defined above, is also used when the sensors are mounted on the train and the readers are at the trackside. The rest of the variables for the trackside deployment case are defined in Table 4.7 and equations (4.2), (4.3), (4.7), and (4.10)–(4.13) show how the variables are computed.

Table 4.7. Trackside Deployment Variables

Variable	Comment
α	Probability of false alarm for sensor.
ϵ	Probability of event detection by sensor.
ρ	Probability of successful communications between trackside reader and sensor.
β	Rate of change of probability of unsuccessful communications with train speed.
η	Probability of unsuccessful communications between trackside reader and sensor when both are stationary.
θ	Real number that specifies the minimum sensor transmission range in meters.
t_{Read}	Real number that states the maximum time in seconds available to read the sensors.
C_H	Acquisition cost of one sensor/seal.
Γ_k	Cost of false alarms per unit.
Δ_k	Cost of missed detection per unit.
Ξ_k	Cost of unsuccessful communications between a trackside reader and the sensors on a unit.
Λ_k	Cost for transmitting messages generated by all the sensors on a rail car.
Ψ_k	Cost of acquiring and installing sensors on each unit.

$$\rho = 1 - (\eta Z + \dot{x}\beta(1 - Z)) \tag{4.10}$$

$$C_H = C_F + \epsilon \text{FP}_2 + \rho \text{FP}_3 + (1 - \alpha)\text{FP}_4 + \theta \text{FP}_5 + \frac{\text{FP}_6}{t_{\text{Read}}} \tag{4.11}$$

$$\Xi_k = \zeta \left(\sum_{\forall\, j,q} \sigma_j y_{jqk} - \sum_{\forall\, i,j,q} \rho \sigma_j S_{ijqk} y_{jqk} \right) \tag{4.12}$$

$$\Lambda_k = \left(\frac{d_T}{\dot{x}} \right) C_c l \sum_{\forall\, i,j,q} \lambda_i S_{ijqk} y_{jqk} \tag{4.13}$$

Suppose that we are given that the probability, ρ, of successful communications from a sensor to a reader varies with train speed according to equation (4.10). In the trackside case the optimization process will determine appropriate values for α, ϵ, ρ (including η and β), θ, and t_{Read}. The cost of acquiring one sensor for the trackside

case is given by equation (4.11). The cost, Ψ_k, of acquiring and installing sensors on each unit in the trackside case is given by substituting equation (4.11) into (4.7). The values for the Γ_k and Δ_k variables are computed using equations (4.2) and (4.3), respectively. As we did above we assume that the likelihood of an event occurring at a container is independent of a sensor detecting that event or the timely notification of that event. In this case we assume that events will get to the operations center in a timely manner if the sensors are read by a trackside reader. The cost of trackside reader failing to read a sensor is given by equation (4.12). This cost is given by the probability of an event times the cost of a trackside reader failing to read a sensor. Equation (4.13) computes the cost of transmitting all the messages generated by all the sensors on a unit. This cost is given by the rail trip duration times the cost of transmitting one message times the message generation rates for all the sensors on a unit.

4.5 Model Descriptions

In this section we present two models for computing the cost metric for a system that uses sensors for cargo monitoring. The models that we develop here are robust enough to handle the following sensor deployment cases:

- A deployment of sensors and a backhaul communications device on the train. This case can be further divided into two subcases:
  - The sensors cannot engage in multihop communications. Instead, they can only communicate with the repeaters or the backhaul communications device. We call this the hierarchical deployment case.
  - The sensors can engage in multihop communications to forward messages

to the backhaul communications device. As a result, this case does not contain any dedicated repeaters. We call this the ad hoc deployment case.

- A deployment of sensors to the train, while the readers and backhaul communications devices are at the trackside. This case can also be split into two subcases for when the train speed is fixed and when it is allowed to vary.

The first model, which is presented in Section 4.5.1, is used when the backhaul communication devices and repeaters are placed on a train. Section 4.5.2 presents the second model, which is used when the train's speed is fixed and backhaul communication devices and readers are placed trackside. Section 4.5.3 shows how the trackside model can be applied in the case where the train speed is allowed to vary. The models discussed in this section are presented using the following general optimization problem formulation:

$$\begin{aligned} &\text{minimize } f_o(x;p) \\ &\text{subject to } f_i(x;p) \leq b_i, \qquad\qquad i = 1, \ldots, m \end{aligned}$$

The objective function, $f_o(x;p)$, will be the system cost metric function, which depends on a vector of variables, x, and a vector of parameters, p. The constraints of the optimization problem are defined by the m f_i equations. When necessary we provide comments relevant to the equations inline.

In Section 4.5.4 we show how the hierarchical sensor deployment can be mapped to the ad hoc sensor deployment case. Our analysis in the next four subsections assumes that the containers on the train are already placed in fixed locations on the train. Thus, Section 4.5.5 briefly mentions an optimization-based approach which can be used to place containers on trains.

4.5.1 Train-mounted Deployment

In this subsection we present a model to minimize the system cost metric of a cargo monitoring system when the sensors and backhaul communications device are on the train. First, we present the objective function and then we discuss the model's constraints, which define valid container and sensor placements.

4.5.1.1 Objective Function for Train-mounted Deployment

Equation (4.14) computes the system cost metric over the duration of a trip:

minimize

$$\sum_{k} \left(\Gamma_k + \Delta_k + \Xi_k + \Lambda_k + \Psi_k + \Upsilon_k + \Omega_k\right) \tag{4.14}$$

The objective function sums the cost of false alarms over a rail journey, cost of missing a detection at a given container, the cost of a sensor failing to communicate in a timely manner, the cost of communications across a rail journey, the material and installation costs of sensors and repeaters, respectively. Finally, the last term in the sum computes the material and installation cost of the backhaul communications device.

4.5.1.2 Constraints for Train-mounted Deployment

The following constraints must be valid for any given optimal deployment of sensors to containers on a train.

subject to

$$\sum_{\forall\, j,q,k} S_{ijqk} \leq 1 \quad \forall\, i \tag{4.15}$$

$$\sum_{\forall\, i,q,k} S_{ijqk} \leq 1 \quad \forall\, j \tag{4.16}$$

Certain attributes (for example, transmission range, detection probability, and

false alarm rate) of the sensors, repeaters, and backhaul communications devices are unique to the network elements. Thus, if a given sensor, for example, is placed on a certain container that same sensor cannot be used on another container. Equation (4.15) ensures that each sensor cannot be simultaneously assigned to more than one container, while equation (4.16) ensures that each container has no more than one sensor.

$$\varphi = \Pr(t \leq \tau) \tag{4.17}$$

$$\sum_{\forall\, i,q,k} \varphi S_{ijqk} \geq \mathrm{TR}_j \qquad \forall\, j \tag{4.18}$$

In equation (4.17) we use the probability distribution defined in Section 3.5.6 to look up the probability of timely notification. Equation (4.18) enforces one of the visibility requirements for container j. In equation (4.18) we require that t, the time taken by a sensor to notify a decision maker of an event, must lie within an interval τ, with probability exceeding some threshold TR_j.

$$\sum_{\forall\, i,q,k} \epsilon S_{ijqk} \geq E_j \qquad \forall\, j \tag{4.19}$$

Equation (4.19) requires that events are detected at container j with a probability ϵ, which exceeds some threshold E_j.

$$\sum_{\forall\, i,q,k} \alpha S_{ijqk} \leq F_j \qquad \forall\, j \tag{4.20}$$

Equation (4.20) enforces the third component of the visibility requirement. In equa-

tion (4.20) we require that the probability of false alarm at container j, α must be kept lower than some threshold F_j. Equations (4.18)–(4.20) ensure that only solutions in the visibility space are considered.

4.5.2 Trackside Deployment with Fixed Train Speeds

In this subsection we present a model to minimize the system cost metric of a cargo monitoring system when the backhaul communications devices and readers are trackside. In this case the train's speed is fixed; however, the probability of successful communications from the sensors to the readers varies with train speed. We intend to study the system trade-offs that exist when monitoring rail-borne cargo. This second model facilitates exploration of the trade-off space by capturing the metrics of a different cargo monitoring methodology, which can be compared with the metrics of the first model. As was done above, the objective function is presented first followed by a discussion of the constraints for this model.

4.5.2.1 Objective Function for Trackside Deployment

Equation (4.21) computes the system cost metric for a trackside-based freight monitoring system over the duration of a trip:

minimize

$$\sum_{k} \left(\Gamma_k + \Delta_k + \Xi_k + \Lambda_k + \Psi_k\right) + \left(\left(\frac{C_A + C_{\mathrm{AD}}}{t_f \times \mathrm{LT}_A} + \frac{C_{\mathrm{BC}} + C_{\mathrm{BD}}}{t_f \times \mathrm{LT}_c}\right) \times \left\lfloor \frac{d_T}{d_A} \right\rfloor\right) \tag{4.21}$$

The sum in the objective function captures the cost of false alarms over a rail journey, the savings that are lost when a sensor either fails to detect that an event has occurred at a container, the cost of communications across a rail journey, and the material and installation costs of sensors. Finally, the last term captures the cost of setting up

trackside readers along a given route.

4.5.2.2 Constraints for Trackside Deployment

Equation (4.15) holds in this case because no sensor can be placed simultaneously on more than one container. We also require that each container can have no more than one sensor, thus, equation (4.16) is also valid in this case. In addition, equations (4.19) and (4.20) are visibility requirements, thus they are also applicable in this case. Finally, the following constraints must also apply:

subject to

$$2\theta - \dot{x}t_{\text{Read}} \geq 0 \tag{4.22}$$

Equation (4.22) says that the minimum time that a sensor is within range of a trackside reader must be greater than the time taken to read a sensor. This constraint allows the train's speed to be limited such that the trackside reader has enough time to read the sensor.

$$2\theta - \dot{x}_{\text{Max}}t_{\text{Read}} \leq 0 \tag{4.23}$$

Equation (4.23) states that sensor view time must be less than or equal to the read time if the train is passing the trackside reader at the maximum speed at which a sensor can be read.

$$d_A \leq \dot{x}\rho\left(\tau - \sum_{r=2}^{5}\tilde{t}_r\right) + 2\theta \tag{4.24}$$

The trackside readers are spaced according to equation (4.24). The r index in equation (4.24) represents the epoch number used in Table 3.4, while $\tilde{t}_r$ represents the

median time for an epoch. Equation (4.24) ensures that the trackside reader separation distance must be less than the distance covered in the event reporting deadline minus the median time taken to notify decision makers of an event.

4.5.3 Trackside Deployment with Variable Train Speeds

In this subsection we present a model to minimize the system cost metric of a cargo monitoring system when the backhaul communications devices and readers are trackside and the train speed can be varied based on sensor parameters. In this case we are optimizing over sensor locations, train speed, and reader separations. This change can be accommodated using equation (4.21) as the objective function.

Constraints: Equations (4.15), (4.16), (4.19), (4.20), (4.22), and (4.24) also hold in this case for the same reasons advanced in Section 4.5.2.2. On the other hand equation (4.23) does not hold since the train speed is not fixed.

4.5.4 Extending the Sensor Placement Models

The presence of repeaters in any system deployment for cargo monitoring adds one more layer of complexity. In Section 4.5 we claimed that a deployment where the sensors can only communicate with repeaters or a backhaul communications device on the train is related to a deployment in which the sensors can engage in multihop communications to forward messages to the backhaul communications device. In this section we discuss how to map the hierarchical deployment case to an ad hoc deployment. In demonstrating this mapping we make the following assumptions:

- The sensor deployment in the hierarchical case is dense enough to have, in the ad hoc case, a fully connected network of sensors with multihop communications capabilities.

- The visibility constraints are the same in all cases and these constraints determine which containers get sensors.
- The probabilities of detection, timely reporting, and false alarm for the sensors do not change as we go from the hierarchical to the ad hoc deployment case.
- Each case contains the same number of backhaul communications devices.
- The ad hoc deployment case does not contain any repeaters.

Suppose that $\mathrm{CM}_{\mathrm{Hier}}$ and $\mathrm{CM}_{\mathrm{AD}}$ represent the cost metrics for the hierarchical and ad hoc deployment cases respectively. Observe that no changes need to be made to the objective function because it simply returns a cost metric when presented with sensor and communications infrastructure locations and their characteristics.

Definitions: Let C_H and C_{HL} represent the acquisition and installation costs for the sensors used in the hierarchical case, while C'_H and C'_{HL} represent the acquisition and installation costs for the sensors used in the ad hoc case. Let J_{Hier} and J_{AD} represent the sets of containers assigned sensors in the hierarchical and ad hoc deployment cases respectively. Let I_{Hier} and I_{AD} represent the sets of communications devices (sensors, repeaters, and backhaul communications) which are assigned in the hierarchical and ad hoc deployment cases, respectively. Furthermore, define S_{Hier} and S_{AD} as the set of sensors in the hierarchical and ad hoc deployment cases. B_{Hier} and B_{AD} and R_{Hier} and R_{AD} are the sets of backhaul communications (B_{XX}) and repeaters (R_{XX}) for the hierarchical and ad hoc deployment cases. Then:

$$I_{\mathrm{Hier}} = S_{\mathrm{Hier}} \cup R_{\mathrm{Hier}} \cup B_{\mathrm{Hier}}$$

$$I_{\mathrm{AD}} = S_{\mathrm{AD}} \cup R_{\mathrm{AD}} \cup B_{\mathrm{AD}}$$

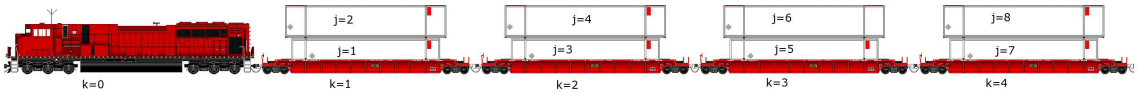


Figure 4.3. Example Train with Sensors Assigned

We claim that, given the assumptions above, the hierarchical sensor deployment case can be mapped to the ad hoc case. This mapping is based on the assumption that sensors which were previously assigned in the hierarchical deployment case are not moved to other containers in the ad hoc case. This mapping is shown in equation (4.25).

$$\begin{aligned}\mathrm{CM}_{\mathrm{AD}} = \mathrm{CM}_{\mathrm{Hier}} &+ C_\alpha \sum_{\substack{i \in I_{\mathrm{AD}} \setminus I_{\mathrm{Hier}} \\ j \in J_{\mathrm{AD}} \setminus J_{\mathrm{Hier}} \\ \forall\, q,k}} \alpha S_{ijqk} y_{jqk} - \zeta \sum_{\substack{i \in I_{\mathrm{AD}} \setminus I_{\mathrm{Hier}} \\ j \in J_{\mathrm{AD}} \setminus J_{\mathrm{Hier}} \\ \forall\, q,k}} \sigma_j S_{ijqk} y_{jqk} \Big(\epsilon + \varphi\Big) \\ &+ D(\Pr(H)C_c + \Pr(I)(1-\Pr(H))C_s)l \sum_{\substack{i \in I_{\mathrm{AD}} \setminus I_{\mathrm{Hier}} \\ j \in J_{\mathrm{AD}} \setminus J_{\mathrm{Hier}} \\ \forall\, q,k}} \lambda_i S_{ijqk} y_{jqk} \\ &+ \sum_{\substack{i \in I_{\mathrm{AD}} \\ j \in J_{\mathrm{AD}} \\ \forall\, q,k}} (C'_H + C'_{\mathrm{HL}}) S_{ijqk} - \sum_{\substack{i \in I_{\mathrm{Hier}} \\ j \in J_{\mathrm{Hier}} \\ \forall\, q,k}} (C_H + C_{\mathrm{HL}}) S_{ijqk} - \sum_{\forall\, q,k} (C_A + C_{\mathrm{AL}}) A_{qk}\end{aligned} \tag{4.25}$$

When proving our claim we will use the example train shown in Fig. 4.3 to illustrate the proof. The train consists of a locomotive and four well cars, with each car bearing two containers. The savings resulting from detecting an event at a container is 8,000 units. The following components are deployed in hierarchical mode for cargo monitoring: a backhaul communications device, a repeater, and seven sensors. The small rectangles on each of the containers in Fig. 4.3 indicate sensor assignments, while a repeater is on container 6 on unit 3, and the backhaul communications device is in the locomotive. Finally, assume that we are given the parameter values shown

Table 4.8. Parameters used in Validating Models

Parameter	Value	Comments
D	20	Rail trip duration in hours.
ζ	0.2	Probability of event occurrence during trip.
F_j	3×10^{-3}	Visibility requirement for probability of false alarm at a container.
E_j	0.85	Visibility requirement for probability of detection at a container.
TR_j	0.85	Visibility requirement for making a timely event report to decision makers.
α	1×10^{-3}	Probability of false alarm for each sensor.
ϵ	0.90	Probability of detection for each sensor.
φ	0.90	Probability of timely event reporting for each sensor.
l	690	Message length in bytes.
λ_i	9.0×10^{-2}	Message generation rate for a sensor. This results in 90 messages every 1,000 hours.
$\Pr(H)$	0.90	Probability of train being in cellular coverage.
$\Pr(I)$	0.90	Probability of train being in satellite coverage.
C_c	5×10^{-5}	Cost in units of sending one byte over a cellular link.
C_s	2×10^{-4}	Cost in units of sending one byte over a satellite link.
$C_{\mathrm{HL}} + C_H$	46	Cost to acquire and install each sensor in the hierarchical case.
$C'_{\mathrm{HL}} + C'_H$	51	Cost to acquire and install each sensor in the ad hoc case.
$C_A + C_{\mathrm{AL}}$	101	Cost to acquire and install each repeater.
C_α	20000	Cost per false alarm.
	14.6	Amortized cost of backhaul communications device.

in Table 4.8.

Proof. 1. If we map the hierarchical sensor deployment case to the ad hoc deployment case, then we must get rid of any repeaters in the deployment (Recall that the ad hoc deployment case does not contain any repeaters.). Therefore, $R_{\mathrm{AD}} = \emptyset$, and any repeaters in the hierarchical case are replaced with sensors in the ad hoc case.

Using Fig. 4.3 as an example we assume, without loss of generality, that sensor 1 is assigned to container 1, sensor 2 is assigned to container 2, etc. Then, in the hierarchical deployment $R_{\mathrm{Hier}} = \{6\}$, i.e., the repeater with index 6 is assigned, while $R_{\mathrm{AD}} = \emptyset$ in the ad hoc case. In addition, assume that in both the hierarchical and ad hoc cases $B_{\mathrm{Hier}} = B_{\mathrm{AD}} = \{9\}$.

2. Since we assume that the same visibility conditions hold in both cases, then we can conclude that the ad hoc deployment case contains at least as many sensors as the hierarchical case, with equality being achieved if the hierarchical case did not contain any repeaters. This condition is captured below:

$$|S_{\text{Hier}}| + |R_{\text{Hier}}| \leq |S_{\text{AD}}| \tag{4.26}$$

Referring to Fig. 4.3, the set of sensors assigned in the hierarchical case is $S_{\text{Hier}} = \{1, 2, 3, 4, 5, 7, 8\}$. The set of sensors assigned in the ad hoc case is $S_{\text{AD}} = \{1, 2, 3, 4, 5, 7, 8, 10\}$. Thus, we see that the claim from equation (4.26) holds with equality.

3. The set of containers that that has sensors in the ad hoc deployment case, but which was not assigned sensors in the hierarchical case is defined as:

$$J_{\text{AD}} \setminus J_{\text{Hier}} \tag{4.27}$$

Observe that this set is empty if no additional containers are assigned sensors in the ad hoc deployment case. Similarly the set of communications devices used in the ad hoc deployment case, but not in the hierarchical case is defined as:

$$I_{\text{AD}} \setminus I_{\text{Hier}} \tag{4.28}$$

As with the containers, this set is empty if no additional communications devices are used in the ad hoc deployment case. Note that, since we assume that both cases contain just one backhaul communications device while the ad hoc

deployment case contains no repeaters, then equation (4.28) simplifies to:

$$S_{\text{AD}} \setminus S_{\text{Hier}} \tag{4.29}$$

Using Fig. 4.3 as an example, then $J_{\text{AD}} \setminus J_{\text{Hier}} = \{6\}$ since container 6 is the only container that has a sensor assigned in the ad hoc deployment case, but which did not have a sensor in the hierarchical deployment. Similarly, $S_{\text{AD}} \setminus S_{\text{Hier}} = \{10\}$, since sensor 10 is the new sensor assigned in the ad hoc case.

4. From equations (4.14) and (4.21) we observe that false alarm and communications costs increase as additional sensors are added, while the savings lost due to missed detections and late event reports decrease. The cost metric of the ad hoc case is the cost metric of the hierarchical case plus the false alarm costs of any new sensors minus the costs of missed detection and untimely reporting due to the new sensors plus any savings from detecting and reporting an event in the desired notification window. To this sum we add the increase in communications costs for the new sensors as well as the installation and material costs for the new sensors. Finally, we subtract the material and installation costs of the repeaters and sensors that were included in the hierarchical deployment. This mapping is summarized in equation (4.25).

   Returning to the example train shown in Fig. 4.3 let us assume that we are given the parameter values in Table 4.8 and that all the containers on the train have low values. Then, the cost metric for the initial hierarchical deployment is 14,178.4 units while the cost metric for the ad hoc deployment is 6,983.5 units. The following costs can be computed for the additional sensor in the ad hoc deployment: false alarm cost for the additional sensor is 20 units, additional

savings in event detection due to the new sensors is 3,600 units, savings resulting from decision maker notification in a timely manner is 3,600 units, the additional communication cost is approximately 0.11 units, cost of acquiring and installing the eight new sensors is 408 units, and the amount gained by not deploying a reader is 101 units. It can be shown that $6983.5 = 14178.4 + 20 - (3600 + 3600) + 0.11 + 408 - 322 - 101$, which confirms equation (4.25).

□

4.5.5 Container Placement

For the purposes of this research we assume that containers have been placed in fixed locations on the train such that the aerodynamic efficiency of the train is maximized. We assume that container placement is done using Lai *et al.*'s [19] method. Please consult [19] for details on the objective function and constraints for this container placement methodology.

4.6 Model Growth and Validation

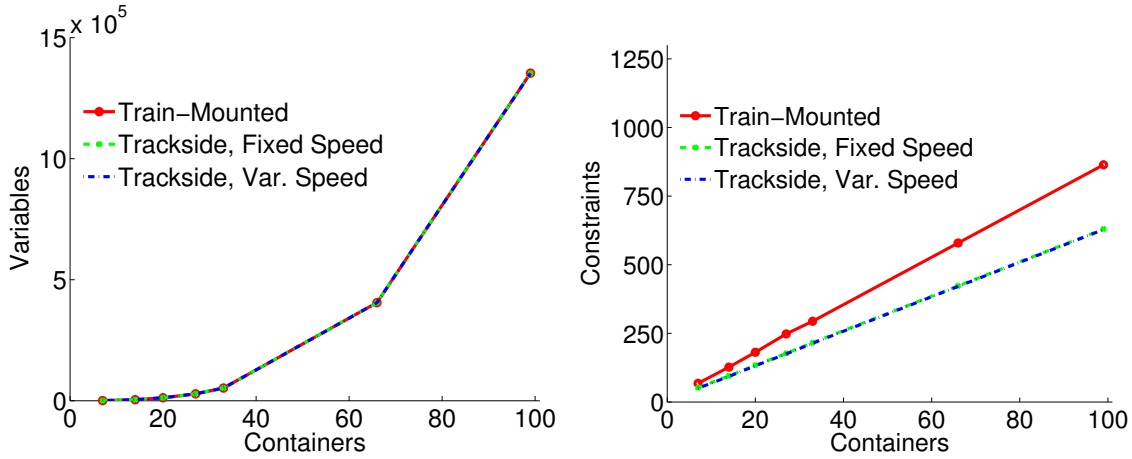
In this section we review model validation and the growth of the sensor placement problem with train size. Model validation seeks to determine if a given mathematical abstraction matches a real system. This task is generally hard to accomplish. Kleindorfer *et al.* [79] provides a more complete discussion on validation of models, especially simulation models. By validating our models we can have greater confidence in the optimization results reported by our models.

4.6.1 Model Growth and Computational Complexity

In this subsection we examine the growth of our models with different problem inputs. The optimization models described in Section 4.5 have been solved using the Bonmin [80] solver running on the NEOS optimization server [81, 82]. Both models have been run for trains with 7, 14, 20, 27, and 33 containers (this translates to 3, 6, 9, 12, and 15 units respectively). The computational complexity of our models depends on the number of variables and constraints, with the problem becoming more complex with more variables and constraints. The growth in the number of variables and constraints is summarized in Fig. 4.4. From Fig. 4.4a it is clear that the train-mounted and trackside models have about the same number of variables. Note that the trackside model with fixed train speeds has additional variables, e.g., sensor transmission range and sensor read time, that are not found in the train mounted model. From Fig. 4.4b we see that the number of constraints in all three models increases gradually with train size. This growth is partially due to the fact that there is one instance of equation (4.15) for every sensor and one instance each of equations (4.16), (4.19), and (4.20) for each container. The rapid growth in the number of variables motivates us to consider using heuristics to assign sensors and related communications infrastructure. In our future work we specify a heuristic for assigning sensors to containers in fixed positions on a train. In the rest of this dissertation we only consider the train-mounted and the trackside model with fixed train speeds.

4.6.2 Model Validation

In this subsection we construct arguments for validating the train-mounted and trackside models by studying trends in the behavior of the optimization models at the



(a) Variables (b) Constraints

Figure 4.4. Problem Growth in Number of Variables and Constraints

Table 4.9. Additional Parameters used in Validating Models

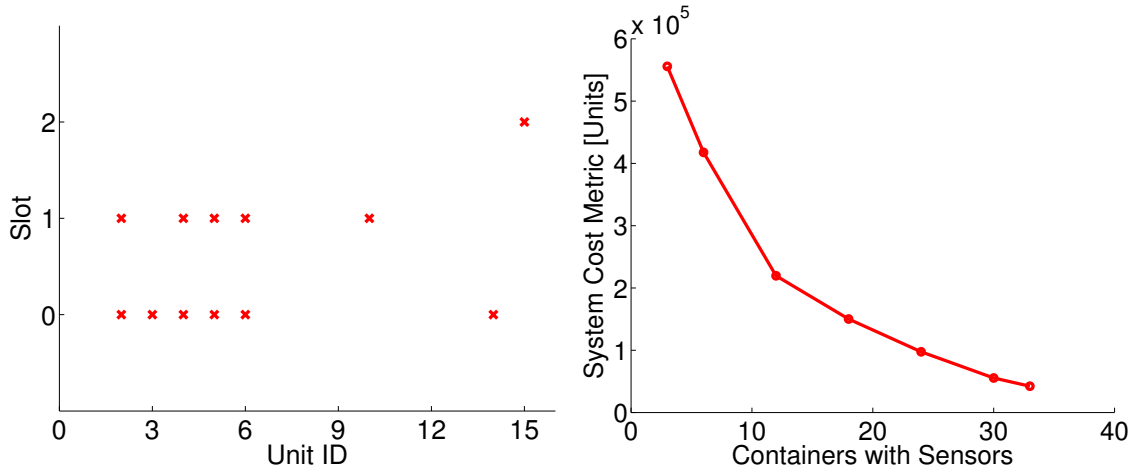
Parameter	Value	Comments
σ_j	$200,000$	Average savings resulting from event detection at high value container. Reference [83] indicates that in 2006 the average container entering the US had a value of 66,000.
σ_j	$100,000$	Average savings resulting from event detection at medium value container.
σ_j	$20,000$	Average savings resulting from event detection at low value container.

boundaries of the visibility space. For the sake of discussion we will use an example train to illustrate our claims. We use the parameter values from Tables 4.8 and 4.9 in our discussion.

4.6.2.1 Train-Mounted Model

Suppose we have a train with 15 units and 33 containers; where 20 of the containers have a low value, 9 have a medium value, and 4 have a high value. If the train-mounted model achieves an optimal result, it returns the cost metric at the optimal solution as well as the final sensor assignment.

Assume that there are initially enough sensors for each of the containers. Suppose



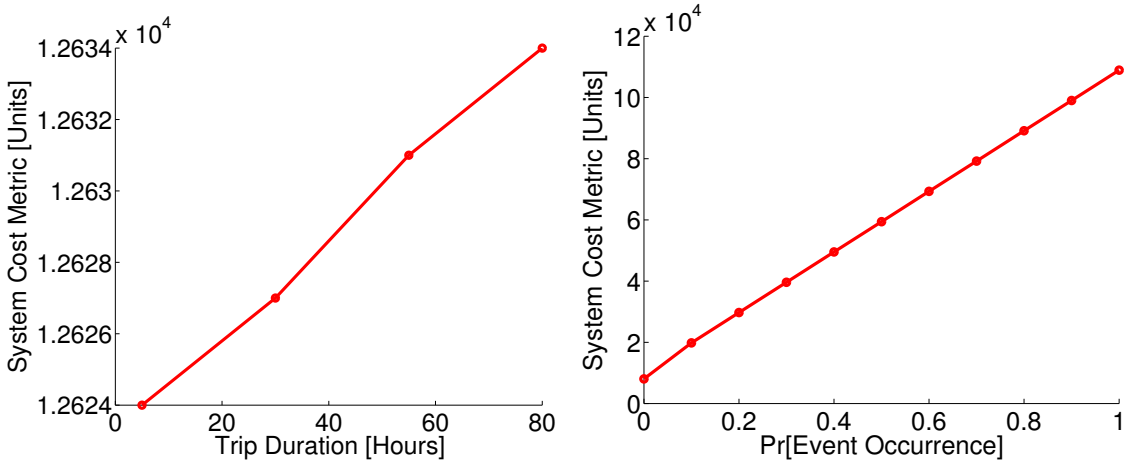
(a) Sensor Locations, where locomotive = unit 0, slots 0 and 2 are in the bottom level of the unit, and slot 1 is in the top level.

(b) Visibility vs. Cost

Figure 4.5. Train-mounted Model: Sensor Locations and Cost Metric Variation with Number of Visible Containers

that the visibility conditions on the containers are relaxed such that: $\mathrm{TR}_j = 0.0, E_j = 0.0$, and $F_j = 1.0$, for some of the containers. In addition assume that there are exactly enough sensors available to satisfy the visibility constraints. Fig. 4.5a shows the slot and unit locations when only 12 of the 33 containers are visible. Fig. 4.5b shows the relationship between the number of visible containers and the cost metric. As we have fewer sensors the cost metric per trip increases as more containers are not "protected" by any sensors.

As the rail trip duration is increased the cost metric per trip should increase as there is greater opportunity for messages to be transmitted. Fig. 4.6a shows that as the rail trip duration is increased the system cost metric also increases. Fig. 4.6b shows the relationship between the probability of event occurrence and the system cost metric. As events become more likely, the system cost metric per trip also increases. Figs. 4.5b and 4.6 show that the train-mounted model exhibits correct trends.



(a) Trip Duration vs. Cost

(b) Prob. of Event Occurrence vs. Cost

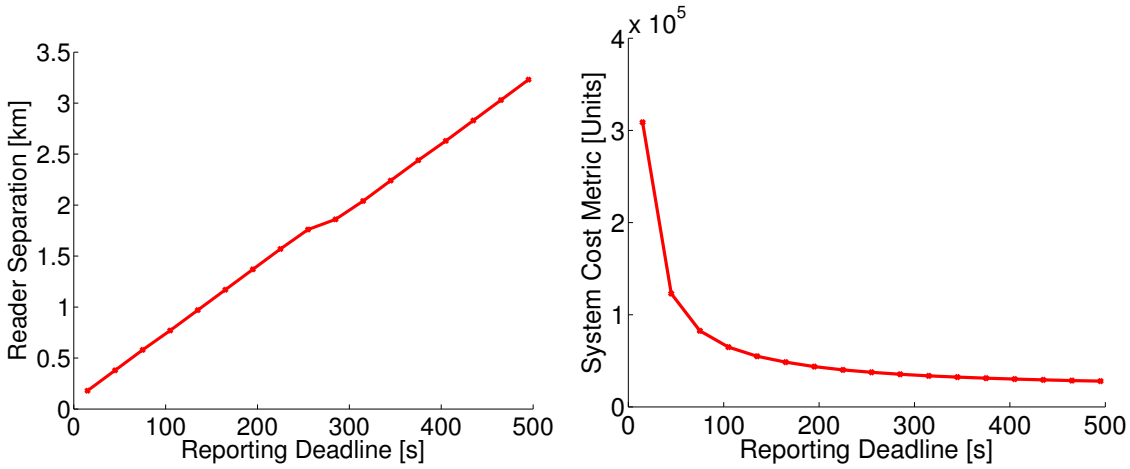
Figure 4.6. Train-mounted Model: Trip Duration and Pr[Event Occurrence] versus Cost Metric

4.6.2.2 Trackside Model with Fixed Speeds

As stated in Sections 4.4.2 and 4.5.2 the outputs of the trackside model include the system cost metric, sensor locations, maximum sensor read time, and minimum sensor transmission range. In addition we also compute reader separation given the reporting deadline and probability of successful communications from a sensor to a trackside reader.

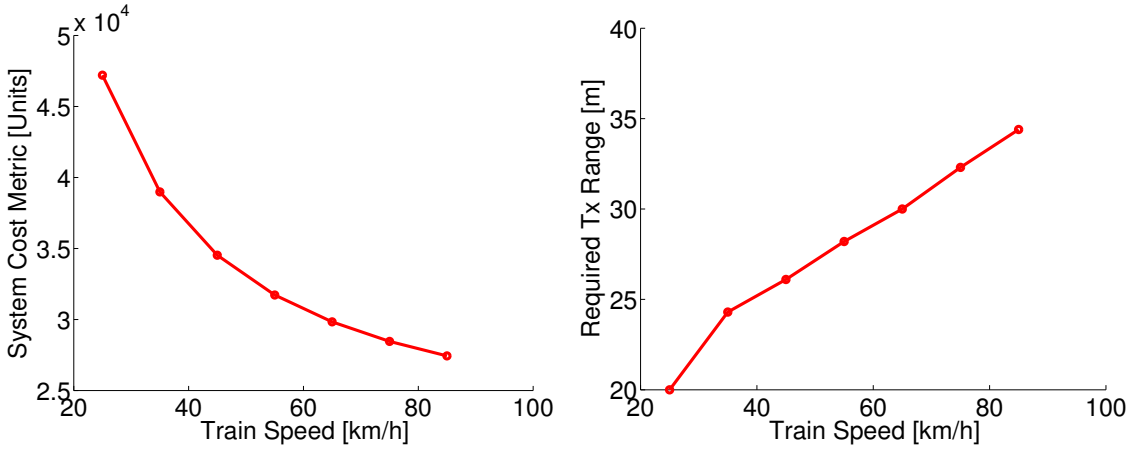
For the trackside model the cost metric for the entire system will increase, as was the case for the train-mounted model, as fewer sensors are available to be used on the train. This is because more of the containers are not protected by sensors. Assume that we have the same train configuration mentioned in Section 4.6.2.1, with each container being assigned a sensor while the readers are at the trackside.

Fig. 4.7 shows the effect of changes in the expected reporting deadline on reader separation and system cost metric when the train speed is fixed at 25 km/h. Fig. 4.7a shows the relationship between reporting deadline and reader separation. As the



(a) Deadline vs. Reader Separation (b) Deadline vs. Cost

Figure 4.7. Trackside Model: Reporting Deadline versus Reader Separation and Cost Metric



(a) Train Speed vs. Cost Metric (b) Train Speed vs. Sensor Transmission Range

Figure 4.8. Trackside Model: Train Speed versus Cost Metric and Sensor Transmission Range

reporting deadline is reduced the trackside readers need to be placed closer together. Since more readers are required, the cost metric increases significantly as the reporting deadline is shortened. Fig. 4.7b shows the change in cost metric with the reporting deadline.

Fig. 4.8a shows that the cost metric decreases as the train speed is increased. As

the train speed is increased the train can cover the distance between its origin and destination in a shorter time implying that the trackside readers can be placed further apart while satisfying the reporting deadlines. Finally, suppose that the system specifications state that each sensor is read in at most 3 s. As the train speed is increased equation (4.22) shows that the sensor transmission range must increase so that each sensor can be read in the specified interval. Fig. 4.8b shows that the sensor transmission range increases as expected. This relatively simple example shows that equation (4.22) correctly captures system operation for the trackside model.

In this section we have shown that our optimization models exhibit correct trends matching a real system. Therefore, we can have confidence in our results.

4.7 Conclusion

This chapter presented two models that can be used to find the optimal cost metric for a rail-borne cargo monitoring system. We presented the parameters and variables for our models. The models presented in Section 4.5 are suitable to enable quantitative evaluation of the trade-offs that can be made when monitoring rail-borne cargo. In addition this chapter has shown that a hierarchical deployment of sensors can be mapped to an ad hoc sensor assignment, given that the sensors assigned in the initial case are not moved to other containers. In Chapter 5 the models developed in this chapter will be used to study possible system trade-offs when seeking visibility into cargo shipments. Finally, this chapter has shown that there is a large number of variables involved in the models for sensor assignment. As a result, Chapter 6 will determine if heuristics can yield near-optimal performance for sensor assignment.

Chapter 5

System Trade-offs and Design of Communications Systems and Networks for Monitoring Cargo in Motion along Trusted Corridors

5.0 Chapter Summary

Due to the high cost of cargo theft, shippers and their customers desire visibility into shipments. We envision that sensors, such as wire seals, and communications devices can be placed on containers and trains to allow customers to know where their cargo is located and what has happened to it. Two mixed integer nonlinear programs have been developed to determine the cost-effectiveness of placing sensors on trains for monitoring cargo, the desired sensor characteristics to provide visibility into shipments as well as the sensor to container mapping. The models enable the study of trade-offs when designing communications systems and networks for monitoring cargo. This chapter provides discussion of those trade-offs as well as a study of the parameters and variables that have the greatest effect on the system cost.

5.1 Introduction

There are several possible system trade-offs that can be made when seeking visibility into cargo in motion on trains. In Chapter 4 we developed two mixed integer nonlinear program (MINLP) models to determine optimal sensor assignments to con-

tainers on a train. Both models have been solved using the Bonmin [80] solver running on the Network-Enabled Optimization System (NEOS) server [81, 82]. The models have been run for trains with 7, 14, 20, 27, and 33 containers (this translates to 3, 6, 9, 12, and 15 units respectively) and it has been shown that it was cost-effective to use sensors to monitor cargo on trains. This optimization formulation yields sensor mappings, i.e., sensor locations, and determines appropriate values for all other variables. The developed models will be used here to determine system trade-offs when seeking visibility into intermodal shipments.

The objective of this chapter is to study trade-offs when monitoring cargo in motion and to identify the important factors that system architects must consider when choosing to implement either a train-mounted or trackside deployment system. Through the use of different sensor cost models the results from this chapter will provide tools for designers of cargo monitoring systems that balance performance and cost. As a result this chapter studies system trade-offs especially at extremes. At one extreme every container could be assigned a sensor and backhaul communications capability. At another extreme only valuable containers could be assigned a sensor and a low cost radio for communications to a single collector and backhaul communications system, and at another extreme there is no backhaul communications capability on the train and trackside readers are deployed. There are system trade-offs in all of these cases, e.g., between system cost and the time needed to report events. First, we study the trade-offs that are evident in the train-mounted system deployment, next, we study the trade-offs when using the trackside system deployment as well as the variables and parameters that have the greatest effect on system cost. Finally, we compare the train-mounted and trackside systems to study trade-offs. For each system the reported cost metric has been minimized for all the cases reported and we

consider the following questions:

- What is the effect of changes in the probability of detection on the system cost metric?
- What is the effect of changes in the probability of timely notification on the system cost metric?
- Suppose that we lack precise knowledge about the probability of detection offered by the sensors, and all we have is an average and a range. What is the effect of these variations on the system cost metric?
- How does the system cost metric vary for different container savings distributions? Given the same container savings distribution how does the system cost metric vary with changes in the probability of event occurrence?

Another objective of this chapter is to highlight the power of the models developed in Chapter 4. To this end this chapter will also identify the system parameters or variables that have the greatest effect on system cost. The rest of this paper is laid out as follows: in Section 5.2 we present the parameter values chosen for exercising our optimization models. Sections 5.3–5.6 examine system trade-offs. Concluding remarks are provided in Section 5.7.

5.2 Sensor Cost Models and Parameter Selection

This section introduces two different sensor cost models as well as the parameter values chosen for exercising the models developed in Chapter 4. The system trade-offs studied in this chapter are dependent on the cost of network elements, particularly sensor cost and mode of communications. In general, the unit sensor cost is a function of the characteristics shown in Table 5.1.

Table 5.1. Sensor Characteristics

Symbol	Comments
α	Sensor's probability of false alarm.
ϵ	Probability of detection offered by sensor.
ρ	Probability of successful communications between sensor and trackside reader.
θ	Minimum separation in kilometers between sensor and reader, i.e., sensor transmission range.
t_{Read}	Maximum time in hours available to read the sensors. This time would typically be measured in seconds, but given to the system designer in hours to ensure that all units are consistent in a model realization.

Four different sensor cost models are studied in this chapter. Two of the sensor cost models apply when there is a linear relationship between the unit sensor cost and the probabilities of detection and false alarm. The rest of the sensor cost models describe the case where there is a nonlinear relationship between the sensor characteristics and the unit sensor cost. There will be two broadly defined classes of sensors—perfect and imperfect sensors. Perfect sensors have probabilities of detection and false alarm of 1 and 0 respectively. Perfect sensors for the trackside deployment case will also have a probability of successful communications with a trackside reader of 1. Imperfect sensors are defined as having a probability of detection less than or equal to 0.975 and a probability of false alarm greater than or equal to 0.001. In practice perfect sensors are unrealizable, thus, the nonlinear sensor cost models studied in this chapter should yield infinite cost if the sensors are perfect. Sensors with probability of detection and false alarm characteristics between perfect and imperfect sensors are defined as Almost perfect sensors. Fig. 5.1 shows the boundaries for the different classes of sensors defined above. The cost models presented in this section map every point in Fig. 5.1 to a sensor cost. Note, that the arrow labelled perfect sensors points to the point where the probability of false alarm is 0 and the probability of detection is 1, whereas the other arrow points to the rectangle shaded in gray.

Assume that each of the sensor characteristics listed in Table 5.1 contributes an

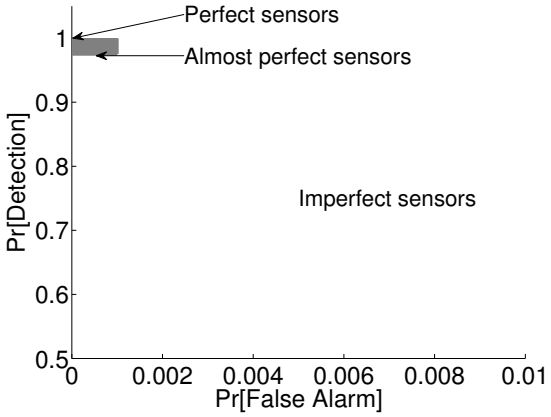


Figure 5.1. Boundaries for Different Sensor Classes

equal amount to the unit sensor cost. The *Linear sensor cost* model for the train-mounted system deployment assumes that the unit sensor cost, C_H^{L}, is computed using equation (5.1).

$$C_H^{\mathrm{L}}(\epsilon, \alpha) = C_F + \epsilon \mathrm{FP}_1 + (1 - \alpha)\mathrm{FP}_2 \tag{5.1}$$

For the train-mounted system deployment the sensors and the readers are stationary relative to one another. Furthermore, assume that a sensor (seal) for the train-mounted system deployment costs 250 units off the shelf [84], and that this sensor has a probability of detection of 0.75 and a 0.005 probability of false alarm. Since the probability of successful communications, ρ, sensor transmission range, θ, and the sensor read time, t_{Read}, are fixed for this system deployment suppose that the fixed portion of the sensor cost, C_F, is 150 units. Using equation (5.1) and the assumption that each sensor characteristic contributes an equal amount to the unit sensor cost, equations (5.2)–(5.4) can also be created.

$$C_H^{\mathrm{L}}(\epsilon = 0, \alpha = 1) = C_F = 150 \tag{5.2}$$

$$C_H^{\mathrm{L}}(\epsilon = 0.75, \alpha = 1) = C_F + 0.75\mathrm{FP}_1 = 200$$
$$\Rightarrow \mathrm{FP}_1 = 66.67 \tag{5.3}$$

$$C_H^{\mathrm{L}}(\epsilon = 0, \alpha = 0.005) = C_F + (1 - 0.005)\mathrm{FP}_2 = 200$$
$$\Rightarrow \mathrm{FP}_2 = 50.25 \tag{5.4}$$

The *Nonlinear sensor cost* model for the train-mounted system captures the fact that perfect sensors have infinite cost and are, therefore, unrealizable. The *Nonlinear sensor cost* model assumes that the unit cost, C_H^{NL}, is computed using equation (5.5).

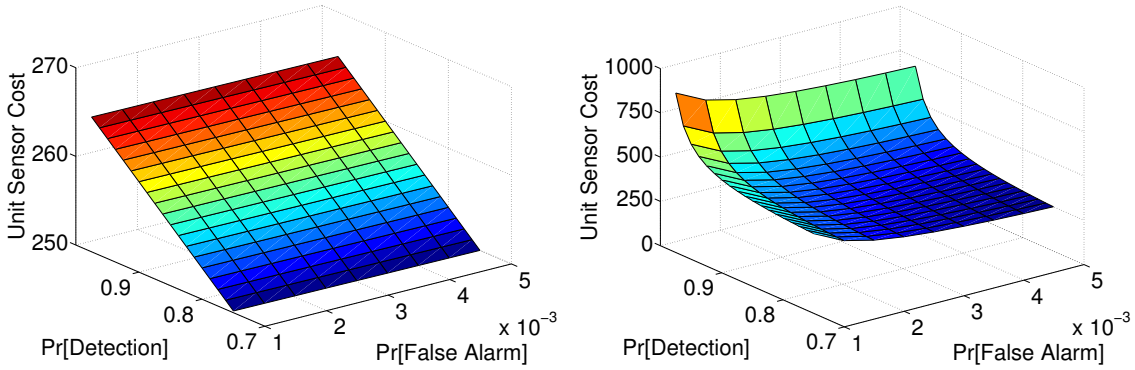
$$C_H^{\mathrm{NL}}(\epsilon, \alpha) = C_F + \frac{\mathrm{FP}_1}{1 - \epsilon} + \frac{\mathrm{FP}_2}{\alpha} \tag{5.5}$$

It is also assumed for the *nonlinear sensor cost* model that a sensor (seal) costs 250 units off the shelf, and that this sensor has a probability of detection of 0.75 and a 0.005 probability of false alarm. Assume that the fixed portion of the sensor cost C_F is also 150 units in this case since the transmission range, θ, sensor read time, t_{Read}, and probability of successful communications, ρ, are fixed for this deployment. Using equation (5.5) and the assumption that each sensor characteristic contributes an equal amount to the unit cost of a sensor off the shelf, then equations (5.6) and (5.7) can also be created.

$$C_H^{\mathrm{NL}}(\epsilon = 0.75, \alpha = 1) = C_F + \frac{\mathrm{FP}_1}{1 - 0.75} + \mathrm{FP}_2$$
$$= 200 + \mathrm{FP}_2 \tag{5.6}$$

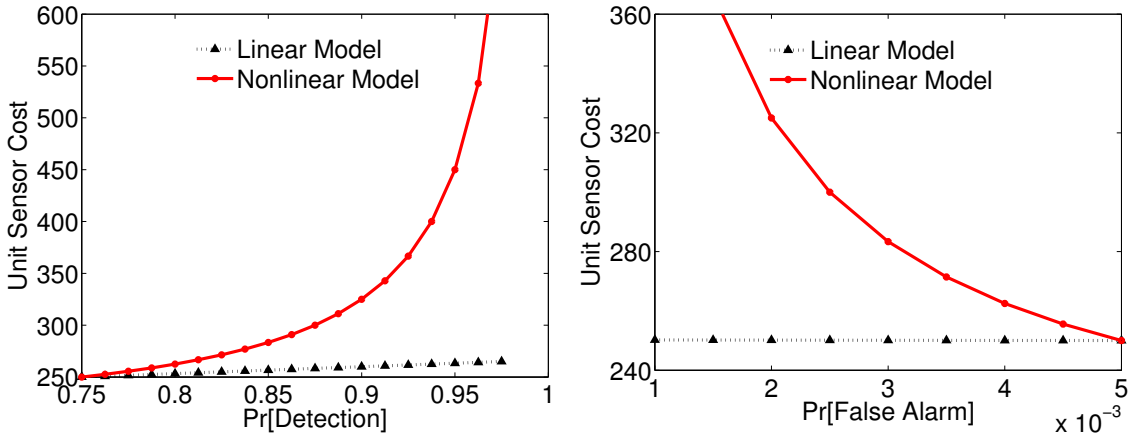
$$C_H^{\mathrm{NL}}(\epsilon = 0, \alpha = 0.005) = C_F + \mathrm{FP}_1 + \frac{\mathrm{FP}_2}{0.005}$$
$$= 200 + \mathrm{FP}_1 \tag{5.7}$$

Solving equations (5.6) and (5.7) FP_1 is found to be 12.50, while $\mathrm{FP}_2 = 0.25$.



(a) Linear Sensor Cost (b) Nonlinear Sensor Cost

Figure 5.2. Sensor Cost Models for Train-Mounted System



(a) $\alpha = 0.005$ (b) $\epsilon = 0.75$

Figure 5.3. Comparison of Linear and Nonlinear Sensor Cost Models

Fig. 5.2 shows how the sensor cost models for the train-mounted system vary with probabilities of detection and false alarm. Fig. 5.3 shows how the unit sensor costs vary for the linear and nonlinear models when the probability of false alarm is fixed at 0.005.

For the trackside deployment system assume that all of the sensor characteristics can be modified at some cost to the system designer. Furthermore, assume that a sensor costs 250 units off the shelf, and that this sensor has a 0.005 probability of false alarm, probability of detection of 0.75, 0.75 probability of successful communications

with a trackside reader, transmission range of 50 m, and a sensor read time of 2 s. As was done above assume that each characteristic contributes an equal amount to the cost of a sensor off the shelf. The unit sensor cost for the *Linear sensor cost* model describes the case where there is a linear relationship between the unit sensor cost and the probabilities of detection, false alarm, and successful communication with a trackside reader. In this case there is a nonlinear relationship between sensor cost and the sensor read time. The *Linear sensor cost*, C_H^{L}, of the trackside deployment system is computed using equation (5.8).

$$C_H^{\mathrm{AL}}(\epsilon, \alpha, \rho, \theta, t_{\mathrm{Read}}) = C_F + \mathrm{FP}_1\epsilon + \mathrm{FP}_2(1-\alpha) + \mathrm{FP}_3\rho + \theta\mathrm{FP}_4 + \frac{\mathrm{FP}_5}{t_{\mathrm{Read}}} \tag{5.8}$$

Assume that the fixed portion of the sensor cost, C_F, is 0 units since no sensor characteristic is fixed, unlike the case for the train-mounted system. Using equation (5.8) and the assumption that each sensor characteristic contributes an equal amount to the unit cost of a sensor off the shelf, then equations (5.9)–(5.13) can be created. In equations (5.9)–(5.13) note that 50 m and 2 s have been converted to kilometers and hours, respectively for model consistency.

$$\begin{aligned} C_H^{\mathrm{AL}}(\epsilon = 0.75, \alpha = 1, \rho = 0, \theta = 0, t_{\mathrm{Read}} = 0.0006) &= 100 \\ 0.75\mathrm{FP}_1 + 1800\mathrm{FP}_5 &= 100 \end{aligned} \tag{5.9}$$

$$\begin{aligned} C_H^{\mathrm{AL}}(\epsilon = 0, \alpha = 0.005, \rho = 0, \theta = 0, t_{\mathrm{Read}} = 0.0006) &= 100 \\ \mathrm{FP}_2(1 - 0.005) + 1800\mathrm{FP}_5 & \end{aligned} \tag{5.10}$$

$$\begin{aligned} C_H^{\mathrm{AL}}(\epsilon = 0, \alpha = 1, \rho = 0.75, \theta = 0, t_{\mathrm{Read}} = 0.0006) &= 100 \\ 0.75\mathrm{FP}_3 + 1800\mathrm{FP}_5 &= 100 \end{aligned} \tag{5.11}$$

$$C_H^{\mathrm{AL}}(\epsilon = 0, \alpha = 1, \rho = 0, \theta = 0.05, t_{\mathrm{Read}} = 0.0006) = 100$$
$$0.05\mathrm{FP}_4 + 1800\mathrm{FP}_5 = 100 \tag{5.12}$$

$$C_H^{\mathrm{AL}}(\epsilon = 0, \alpha = 1, \rho = 0, \theta = 0, t_{\mathrm{Read}} = 0.0006) = 50$$
$$1800\mathrm{FP}_5 = 50 \tag{5.13}$$

Solving equations (5.9)–(5.13) FP_1 is found to be 66.67, $\mathrm{FP}_2 = 50.25$, $\mathrm{FP}_3 = 66.67$, $\mathrm{FP}_4 = 1000$, and $\mathrm{FP}_5 = 0.028$.

The unit sensor cost for the *Nonlinear sensor cost* model of the trackside deployment system captures the case that perfect sensors are unrealizable, as was the case for the train-mounted system. With the *Nonlinear sensor cost* model the unit sensor cost, C_H^{L}, is computed using equation (5.14).

$$C_H^{\mathrm{NL}}(\epsilon, \alpha, \rho, \theta, t_{\mathrm{Read}}) = C_F + \frac{\mathrm{FP}_1}{1-\epsilon} + \frac{\mathrm{FP}_2}{\alpha} + \frac{\mathrm{FP}_3}{1-\rho} + e^{\theta \mathrm{FP}_4} + \frac{\mathrm{FP}_5}{t_{\mathrm{Read}}} \tag{5.14}$$

As was done for the *linear sensor cost* assume that the fixed portion of the sensor cost, C_F, is 0 units since no sensor characteristic is fixed. Assume that a sensor costs 250 units off the shelf, and that this sensor has a 0.005 probability of false alarm, probability of detection of 0.75, 0.75 probability of successful communications with a trackside reader, transmission range of 50 m, and a sensor read time of 2 s. Equation (5.15) shows the unit sensor cost computation for this initial case.

$$C_H^{\mathrm{NL}}(\epsilon = 0.75, \alpha = 0.005, \rho = 0.75, \theta = 0.05, t_{\mathrm{Read}} = 0.0006) =$$
$$C_F + \frac{\mathrm{FP}_1}{1-0.75} + \frac{\mathrm{FP}_2}{0.005} + \frac{\mathrm{FP}_3}{1-0.75} + e^{0.05\mathrm{FP}_4} + 1800\mathrm{FP}_5 = 250 \tag{5.15}$$

Assuming that each sensor characteristic contributes an equal amount to the unit cost

of a sensor off the shelf, then the values for FP_1–FP_5 can be computed as follows:

$$\frac{FP_1}{1-0.75} = \frac{250}{5} \Rightarrow FP_1 = 12.50 \tag{5.16}$$

$$\frac{FP_2}{0.005} = \frac{250}{5} \Rightarrow FP_2 = 0.25 \tag{5.17}$$

$$\frac{FP_3}{1-0.75} = \frac{250}{5} \Rightarrow FP_3 = 12.50 \tag{5.18}$$

$$e^{0.05FP_4} = \frac{250}{5} \Rightarrow FP_4 = 78.24 \tag{5.19}$$

$$1800FP_5 = \frac{250}{5} \Rightarrow FP_5 = 0.028 \tag{5.20}$$

Due to the assumptions that a sensor costs 250 units off the shelf for both the train-mounted and trackside systems, and that each sensor characteristic contributed an equal amount to the cost of a basic sensor the equations and parameters presented in equations (5.1) and (5.14) enable fair comparisons between the models. In Section 5.3 we use equations (5.1) and (5.5) when studying the trade-offs for the train-mounted system. Equations (5.8) and (5.14) are used in Section 5.4 when studying the trade-offs for the trackside deployment system.

Tables 5.2 and 5.3 present the parameter values chosen for exercising our models. In creating Tables 5.2 and 5.3 we have sought to use observations obtained from the short-haul [49] and long-haul [85] rail trials, which were real deployments of a cargo monitoring sensor network, making the results reported here more credible. Other parameter values presented in Tables 5.2 and 5.3 are chosen by the authors and varied when carrying out sensitivity analysis. Unless otherwise stated the rest of the parameters are fixed.

Table 5.2. Parameters used in exercising models

Parameter	Value	Comments
TR_j	0.80	Visibility requirement for probability of getting a timely report.
τ_j	10–900 s	Deadline in seconds for receiving an event report.
E_j	0.80	Visibility requirement for probability of detection at a container.
F_j	5×10^{-3}	Visibility requirement for probability of false alarm at a container.
$\dot{x}$	25 km/h	Average train speed which is computed from [86, 87]. This parameter was varied from 25–90 in sensitivity analysis.
Z	0.36	Probability of the train being stationary which is computed from [86].
λ_i	9.0×10^{-1} msgs/hr	Number of messages generated per sensor each hour.
ζ	0.0031	Probability of occurrence for critical events at any container on the train; author selected. This parameter was varied from 0.0031–0.1 when carrying out sensitivity analysis on probability of event occurrence.
σ_j	67000 units	Reference [83] indicates that in 2006 the average container entering the U.S. had a value of 66,000.
D	80 hrs	Rail trip duration in hours; based on [86].
d_T	1984 km	Rail trip length, based on distance from Laredo to Kansas City [87].

5.3 Trade-offs and Sensitivity Analysis for Train-Mounted System Deployment

In this section we present the system trade-offs for a train-mounted system deployment. We also discuss the train-mounted deployment system's sensitivity to changes in the values of some of the parameters and variables. In general sensitivity analysis is carried out by taking the partial derivative of the system function with respect to some parameter [90]. Unless otherwise stated, the data in Sections 5.3 and 5.4 was generated under the following assumptions:

- The average train speed was 25 km/h, which is computed from [86, 87].
- The length of the rail trip was 1984 km, which is the distance from Laredo to Kansas City [87].

Table 5.3. Parameters used in exercising models: Cont'd

Parameter	Value	Comments
C_α	20000 units	Cost per false alarm; author selected.
C_c	1.9×10^{-6} units/byte	Cost of transmitting data over cellular link [88].
C_s	5.1×10^{-5} units/byte	Cost of transmitting data over satellite link [89].
C_{HL}	1 unit	Cost of installing one sensor; author selected.
C_A	100 units	Cost of one reader; author selected.
C_{AL}	1 unit	Cost of installing one train-mounted reader.
C_{BC}	400 units	Cost of one backhaul cellular device.
C_{BS}	1000 units	Cost of one backhaul satellite device.
C_{AD}	3000 units	Cost of installing one trackside reader. This was varied from 3,000–102,000 when carrying out sensitivity analysis on trackside reader costs. Note that Ouyang *et al.* [20] state that some trackside detection equipment can cost up to $100,000 or more.
t_L	1.0 /week	Trips per locomotive.
t_F	1.0 /week	Number of locomotives going past a reader.
LT_C	104 weeks	Useful lifetime of cellular modem.
LT_S	104 weeks	Useful lifetime of satellite modem.
LT_A	104 weeks	Useful lifetime of seal reader.
l	690 bytes	Average message length from train to operations center. Based on short-haul trial [49].
RTT_C	3.88 s	Round trip time from train to operations center over cellular link. Based on short-haul trial [49].
RTT_S	8.37 s	Round trip time from train to operations center over satellite link. Based on long-haul trial [85].
$\Pr(H)$	0.80	Probability of train being in cellular coverage. This value was varied from 0.0–1.0 when carrying out sensitivity analysis on the mode of communication.
$\Pr(I)$	0.90	Probability of train being in satellite coverage. This value was varied from 0.0–1.0 when carrying out sensitivity analysis on the mode of communication.

- There were 33 sensors and 33 containers, where 21 containers had a mean value of 20,000 units, 9 containers had a mean value of 100,000 units, and 3 containers had a mean value of 200,000 units.

- Each sensor was assumed to have a transmission range of 50 m, so a repeater was placed on every third railcar in the train to allow the formation of a sensor network through the train.

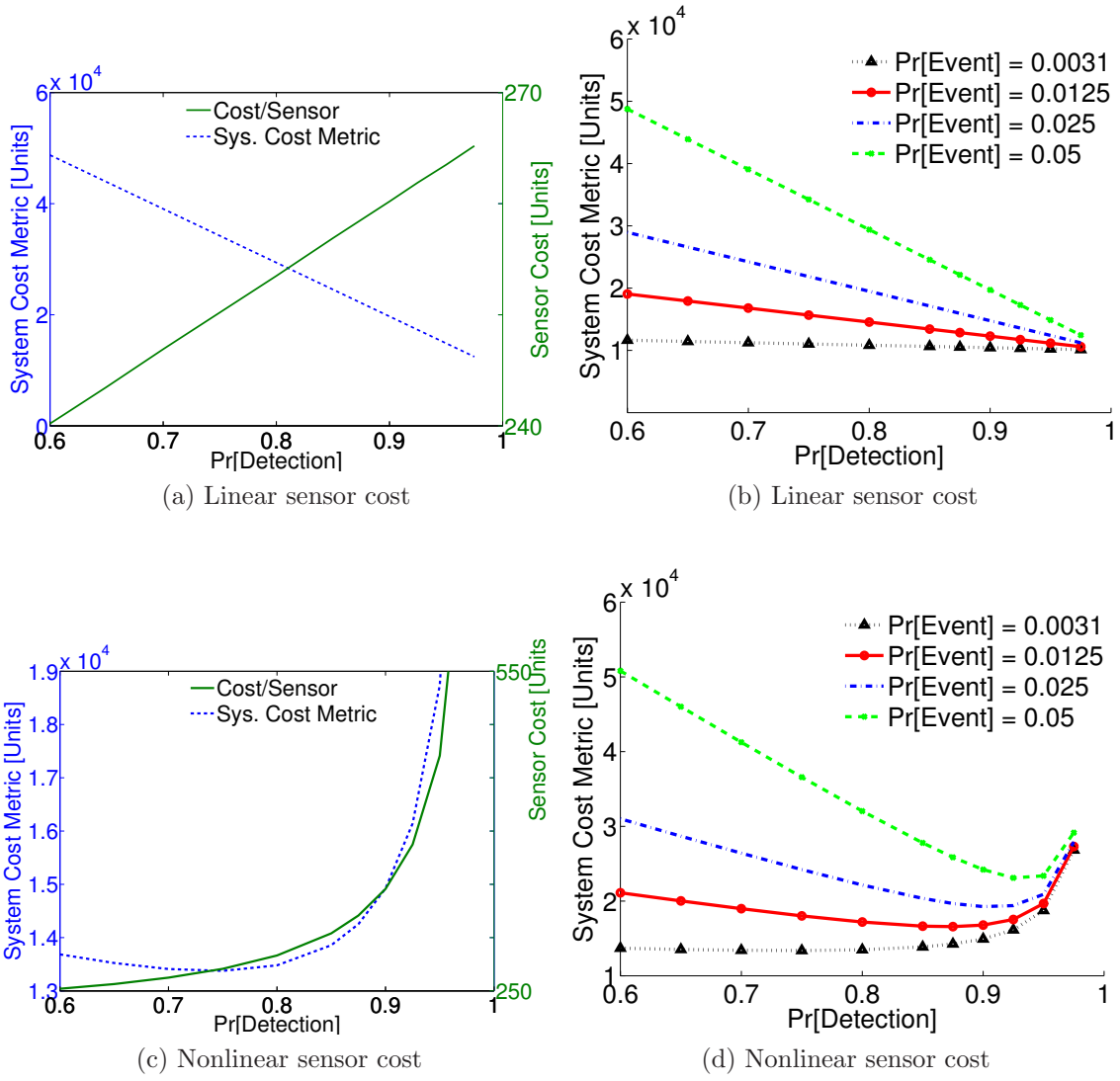
- The probability of a critical event, such as a container seal being opened, closed,

or tampered with, occurring at each container was 0.0031. As a result the probability of a critical event occurring on a train carrying 33 containers is 0.098.

- The deadline for event notification was 5 minutes. For the trackside system deployment this meant readers could be placed between 1.94 km and 6.53 km apart depending on the system deployment parameters and variables.

5.3.1 Trade-offs and Sensitivity Analysis with Probability of Detection

The following questions are considered for the train-mounted system: What is the effect of changes in the probability of detection on the system cost metric? For the *linear sensor cost model* we observe from Fig. 5.4a that the system cost metric decreases as the probability of detection offered by the sensors improves. Thus, it is better for system designers to purchase better quality sensors with a high probability of detection as this ultimately reduces the system cost metric. From Fig. 5.4b we observe that the system cost metric increases, as expected, as the probability of a critical event occurring at a container rises. For the *nonlinear sensor cost model* we observe from Fig. 5.4c that the system cost metric decreases as the probability of detection offered by the sensors is increased to 0.742. Beyond this point the unit cost of the sensors and the system cost metric increase showing that it is not cost-effective to use very expensive sensors with a high probability of detection. Fig. 5.4d is similar to Fig. 5.4b in that the system cost metric increases, as expected, as the probability of a critical event occurring at a container rises. In this case we see that the optimum probability of detection decreases as the probability of event occurrence is reduced; the optimal probability of detection decreases from 0.935 to 0.742 as the probability of event occurrence is reduced from 0.05 to 0.0031. Thus, as the probability of event



(a) Linear sensor cost

(b) Linear sensor cost

(c) Nonlinear sensor cost

(d) Nonlinear sensor cost

Figure 5.4. Train-mounted System: Cost Metric Variation with Probability of Detection

occurrence is reduced it is more cost-effective to protect the cargo with cheaper sensors having a lower probability of detection.

Suppose the system cost metric is represented by the variable μ, then for the linear sensor cost model the analytical absolute sensitivity function [90, p. 9] with respect to the probability of detection is given by equation (5.21). From equation (5.21) we see that for a given train configuration the system cost metric's sensitivity to the

Table 5.4. Train-Mounted System: Sensitivity Function with respect to Probability of Detection

	Linear	Nonlinear	
Pr[Critical Event]	$\frac{\partial \mu}{\partial \epsilon}$	$\frac{\partial \mu}{\partial \epsilon}\Big\vert_{\epsilon=0.75}$	$\frac{\partial \mu}{\partial \epsilon}\Big\vert_{\epsilon=0.80}$
0.0125	-2.255×10^4	-1.815×10^4	-1.444×10^4
0.025	-4.730×10^4	-4.290×10^4	-3.918×10^4
0.05	-9.680×10^4	-9.240×10^4	-8.868×10^4

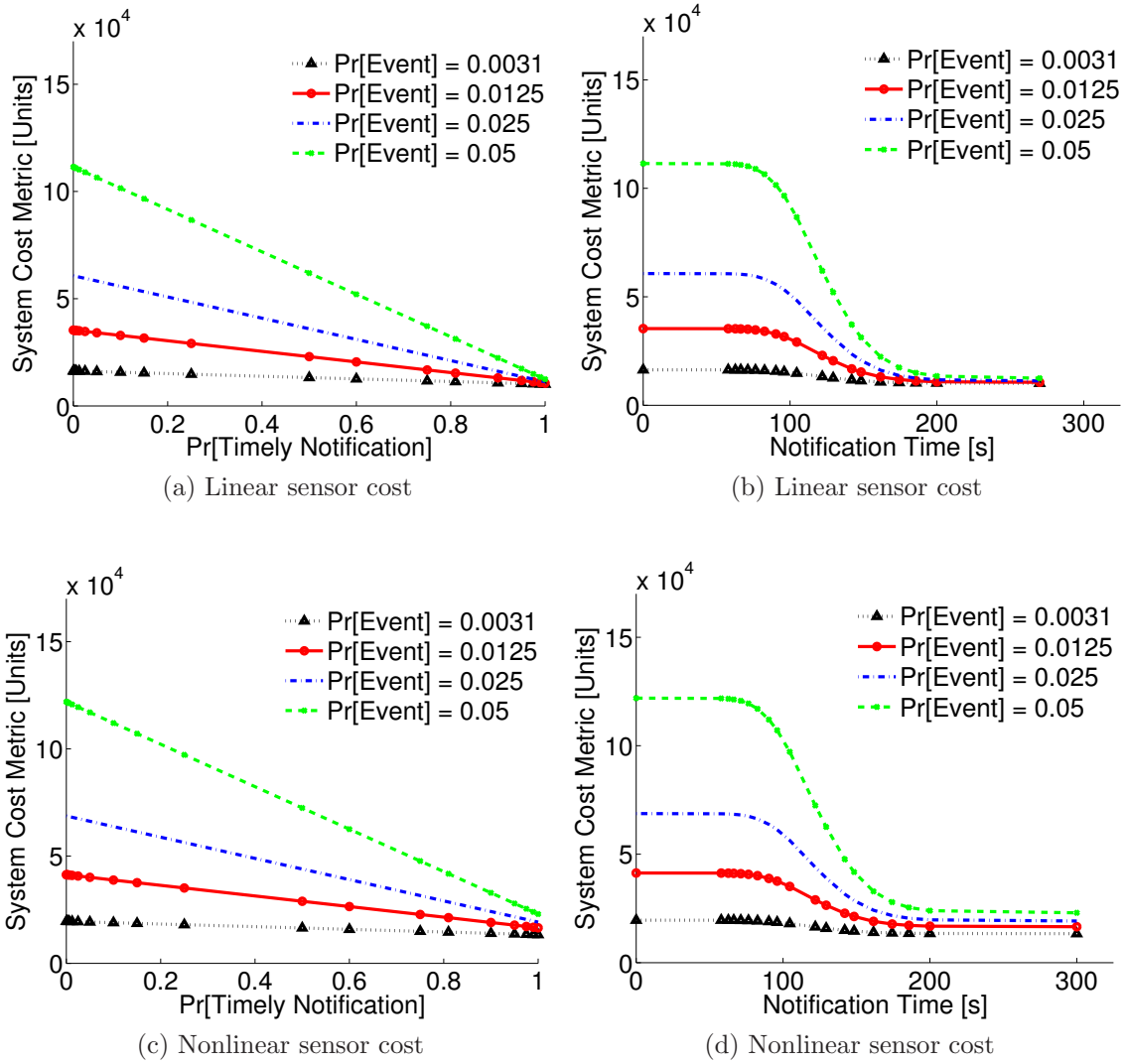
probability of detection is constant as seen in Fig. 5.4a.

$$\frac{\partial \mu}{\partial \epsilon} = \sum_{\forall\, i,j,q,k} \left(\mathrm{FP}_1 - \zeta \sigma_j \right) S_{ijqk} y_{jqk} \tag{5.21}$$

For the nonlinear sensor cost model the absolute sensitivity function with respect to the probability of detection is given by equation (5.22). From equation (5.22) the system cost metric has nonlinear sensitivity to the probability of detection, confirming the observations drawn from Fig. 5.4c. Table 5.4 shows the absolute sensitivity function with respect to probability of detection for both the linear and nonlinear sensor cost models.

$$\frac{\partial \mu}{\partial \epsilon} = \sum_{\forall\, i,j,q,k} \left(\frac{\mathrm{FP}_1}{(1-\epsilon)^2} - \zeta \sigma_j \right) S_{ijqk} y_{jqk} \tag{5.22}$$

From the sensitivity analysis we observe that the nonlinear sensor cost model is marginally less sensitive to the probability of detection, provided the probability of detection is less than the optimal value. For probabilities of detection greater than the optimal value, the nonlinear sensor cost model is more sensitive to probability of detection.



(a) Linear sensor cost

(b) Linear sensor cost

(c) Nonlinear sensor cost

(d) Nonlinear sensor cost

Figure 5.5. Train-mounted System: Cost Metric Variation with Probability of Timely Notification and Notification Time

5.3.2 Trade-offs and Sensitivity Analysis with Probability of Timely Notification

Next, the following questions are considered for the train-mounted system: What is the effect of changes in the probability of timely notification? What are the effects of changes in the probability of event occurrence? What is the effect of the time taken to notify decision makers on the system cost metric? Figs. 5.5a and 5.5c show that

Table 5.5. Train-Mounted System: Sensitivity Function with respect to Probability of Timely Reporting

	Linear	Nonlinear
Pr[Critical Event]	$\frac{\partial \mu}{\partial \varphi}$	$\frac{\partial \mu}{\partial \varphi}$
0.0125	-2.475×10^4	-2.475×10^4
0.025	-4.950×10^4	-4.950×10^4
0.05	-9.899×10^4	-9.899×10^4

the system cost metric decreases as the probability of timely notification increases; however, the system cost metric is higher for the nonlinear sensor cost model due to the marginally higher sensor costs. Since we assumed in Chapter 4 that the impact of the sensors on notification time is negligible the sensor cost does not change with probability of timely notification. From Figs. 5.5b and 5.5d we see that the system cost metric increases, as expected, as the probability of a critical event occurring at a container rises.

The absolute sensitivity function with respect to timely reporting for both the linear and nonlinear sensor cost models is given by equation (5.23). This function is constant for a given train configuration, thus confirming the observation from Figs. 5.5a and 5.5c showing that the slope of the system cost metric against probability of timely reporting is constant. Table 5.5 shows the system cost metric sensitivity with respect to the probability of timely reporting.

$$\frac{\partial \mu}{\partial \varphi} = -\zeta \sum_{\forall i,j,q,k} \sigma_j S_{ijqk} y_{jqk} \tag{5.23}$$

In Section 3.5.6, we used a result from [76] to compute the probability density function (pdf) for timely notification from a sum of independent gamma random variables. Suppose that the pdf is given by f_T, then the cummulative distribution

function for timely notification within an interval, τ, is given by equation (5.24)

$$F_T(\tau) = \int_0^{\tau} f_T(t)dt \tag{5.24}$$

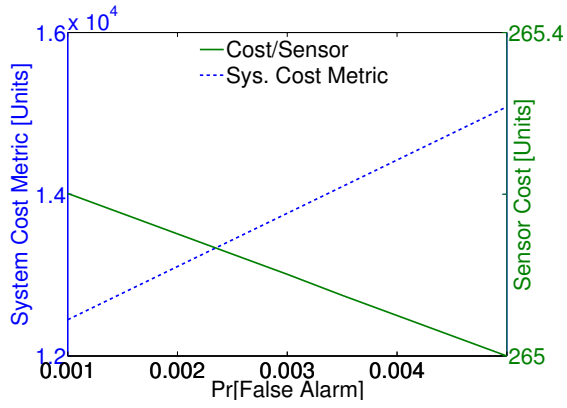
and the notification time, t, associated with a probability, φ, is given by equation (5.25), where $F_T^{-1}(x)$ is the inverse function of $F_T(x)$.

$$t = F_T^{-1}(\varphi) \tag{5.25}$$

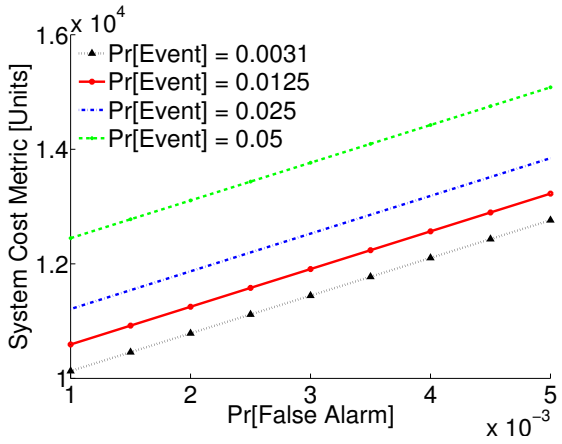
Therefore, every probability of timely notification in Figs. 5.5a and 5.5c is related to a notification time. The higher the notification time, the higher the probability that a decision maker will be notified in the specified time with a corresponding decrease in the system cost metric, as shown in Figs. 5.5b and 5.5d.

5.3.3 Trade-offs and Sensitivity Analysis with Probability of False Alarm

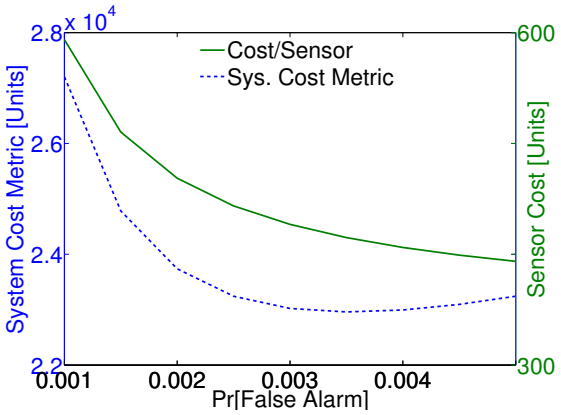
Next, we study the following research questions: What is the effect of changes in the probability of false alarm on the system cost metric? For the linear sensor cost model from Fig. 5.6a we observe that the system cost metric increases, as expected, as the sensors have a higher probability of false alarm. Simultaneously the sensors become cheaper as their probability of false alarm increases. However, a less than 1% decrease in the unit sensor cost leads to an approximately 21.2% increase in the system cost metric. Thus, it is better to invest in expensive sensors that have a low probability of false alarm. Fig. 5.6b shows that the system cost metric rises with the probability of critical event occurrence. Furthermore, this growth occurs at the same rate no matter the probability of critical event occurrence. Ignoring the signs of the slopes of the cost metric in Figs. 5.4a, 5.5a, and 5.6a when the probability of event



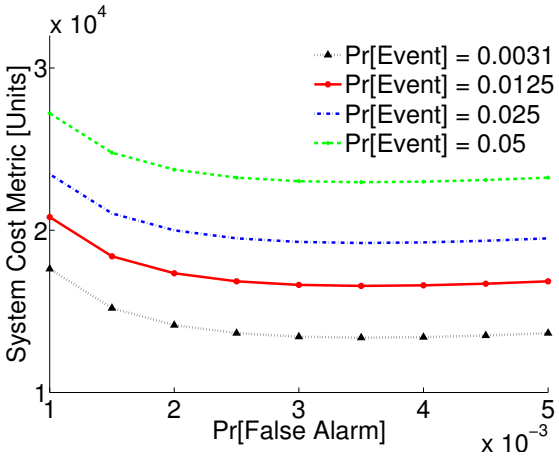
(a) Linear sensor cost



(b) Linear sensor cost



(c) Nonlinear sensor cost



(d) Nonlinear sensor cost

Figure 5.6. Train-mounted System: Cost Metric Variation with Prob. of False Alarm

occurrence is 0.05, then the system cost metric is more sensitive to the probability of false alarm, than to probability of timely reporting or probability of detection. Thus, it is better to invest in sensors with lower probabilities of false alarm and then higher probabilities of detection.

For the nonlinear sensor cost model we observe from Fig. 5.6c that the system cost metric decreases as the probability of false alarm offered by the sensors increases to 0.0035. Beyond this point the system cost metric increases as the higher probability

of false alarm causes rises in the system cost metric. From Fig. 5.6d this trend is repeated for different probabilities of event occurrence. In addition, the system cost metric rises, as expected, as the probability of event occurrence goes up.

For the linear sensor cost model the absolute sensitivity function with respect to the probability of false alarm is given by equation (5.26). From equation (5.26) we see that for a given train configuration the system cost metric's sensitivity to the probability of false alarm is constant as seen in Fig. 5.6a. In addition, this sensitivity is independent of the probability of event occurrence.

$$\frac{\partial \mu}{\partial \alpha} = \sum_{\forall i,j,q,k} \left(C_\alpha - \mathrm{FP}_2 \right) S_{ijqk} y_{jqk} \tag{5.26}$$

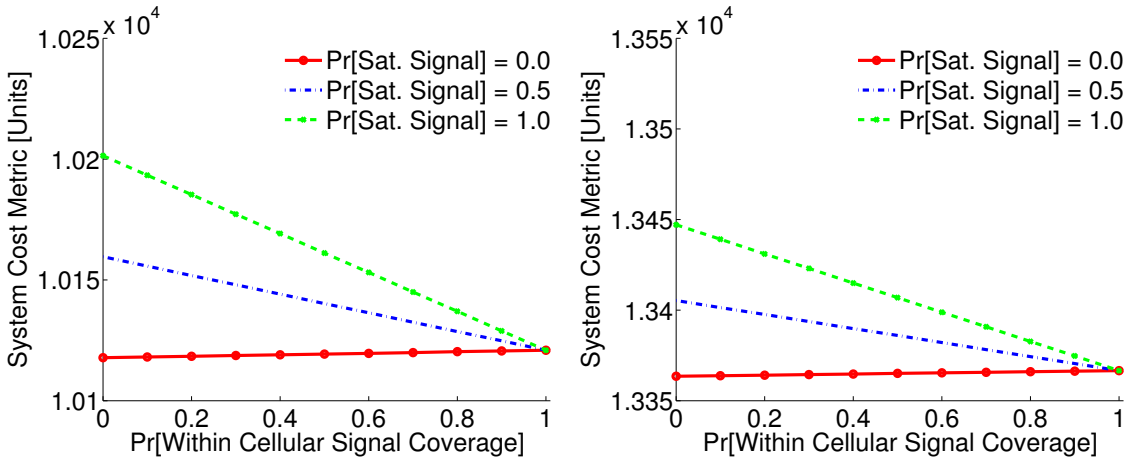
For the nonlinear sensor cost model the absolute sensitivity function with respect to the probability of false alarm is given by equation (5.27). From equation (5.27) the system cost metric has nonlinear sensitivity to the probability of false alarm, confirming the observations drawn from Fig. 5.6c.

$$\frac{\partial \mu}{\partial \alpha} = \sum_{\forall i,j,q,k} \left(C_\alpha - \frac{\mathrm{FP}_2}{\alpha^2} \right) S_{ijqk} y_{jqk} \tag{5.27}$$

Table 5.6 shows the sensitivity of the system cost metric to the probability of false alarm for the linear and nonlinear sensor cost models. Table 5.6 shows that the system with a nonlinear sensor cost model shows less sensitivity to the probability of false alarm if the selected sensors have a probability of false alarm close to the optimal value of 0.0035.

Table 5.6. Train-Mounted System: Sensitivity Function with respect to Probability of False Alarm

	Linear	Nonlinear	
Pr[Critical Event]	$\frac{\partial \mu}{\partial \alpha}$	$\frac{\partial \mu}{\partial \alpha}\Big\|_{\alpha=0.002}$	$\frac{\partial \mu}{\partial \alpha}\Big\|_{\alpha=0.004}$
0.0125	6.583×10^5	-1.403×10^6	1.444×10^5
0.025	6.583×10^5	-1.403×10^6	1.444×10^5
0.05	6.583×10^5	-1.403×10^6	1.444×10^5



(a) Linear sensor cost

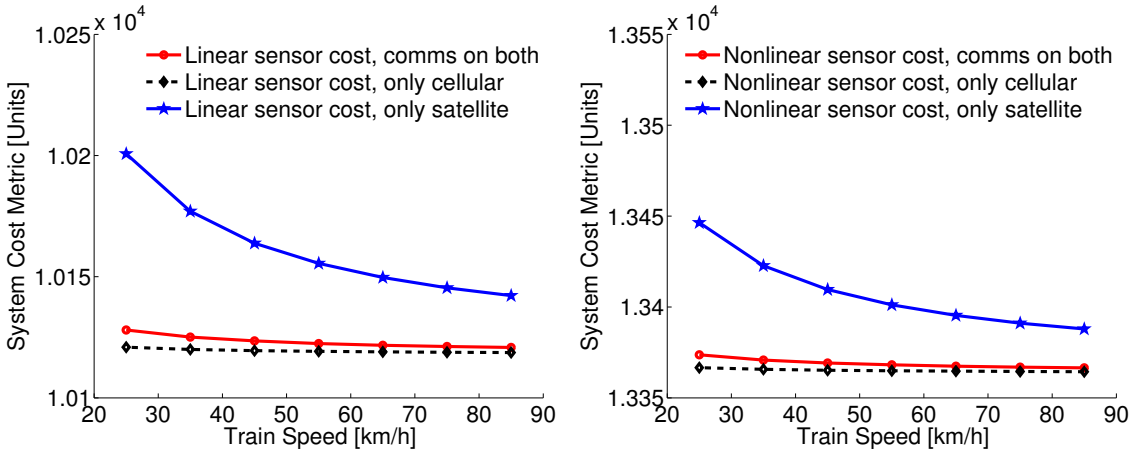
(b) Nonlinear sensor cost

Figure 5.7. Train-mounted System: Cost Metric Variation with Mode of Communications

5.3.4 Trade-offs and Sensitivity Analysis with Train Speed

In this section we use analysis to demonstrate the power of the models developed in Chapter 4 by studying the effects of train speed on the system cost metric. Note that train speed only affects communications costs in the train-mounted system and these communications costs ultimately constitute a very small proportion of the system cost metric. This section will also show how the system cost metric is affected by changes in the mode of communications used by the train.

As we saw in Table 5.3 it costs more to send a message over a satellite link versus over a cellular link. From Figs. 5.7a and 5.7b we see that as the probability of being within cellular signal coverage increases, the system cost metric decreases, except in



(a) Linear sensor cost

(b) Nonlinear sensor cost

Figure 5.8. Train-mounted System: Cost Metric Variation with Mode of Communications and Train Speed

the case where there is no satellite link. In this case the cost metric increases because of increasing communications costs. Furthermore, the system with a nonlinear sensor cost model has a higher system cost metric due to the more expensive sensors in this case. From Fig. 5.8a and 5.8b we see that the system cost metric decreases most with train speed if the train is using a satellite link exclusively for communications; however, this is approximately a 0.4% or 0.25% decrease in the system cost metric for the linear and nonlinear sensor cost models, respectively, as the train speed is increased from 25 to 85 km/h. On the other hand the system cost metric does not change very much with train speed if a cellular link, which is the cheapest communications mode, is used exclusively for communications. If, on the other hand, we had a 90% chance of being in satellite and/or cellular coverage, then there is a 0.05% and 0.03% change in the system cost metric for the linear and nonlinear sensor cost models, respectively. This change in cost is very small and it shows that for a fixed probability of event occurrence and mode of communications, the system cost metric does not change very much with train speed.

Table 5.7. Train-Mounted System: Sensitivity Function with respect to Train Speed

Mode of communications	Linear		Nonlinear	
	$\frac{\partial \mu}{\partial \dot{x}}\Big\vert_{\dot{x}=25}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\vert_{\dot{x}=75}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\vert_{\dot{x}=25}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\vert_{\dot{x}=75}$
Only cellular	-0.124	-0.014	-0.124	-0.014
Only satellite	-3.318	-0.369	-3.318	-0.369
Comms. on both	-0.410	-0.046	-0.410	-0.046

For both the linear and nonlinear sensor cost models the absolute sensitivity function with respect to train speed is given by equation (5.28). From equation (5.28) we see that for fixed message generation rate, message length, and probabilities of being in cellular and satellite coverage, the system cost becomes less sensitive to train speed as the train travels faster. This observation confirms the results shown in Fig. 5.7.

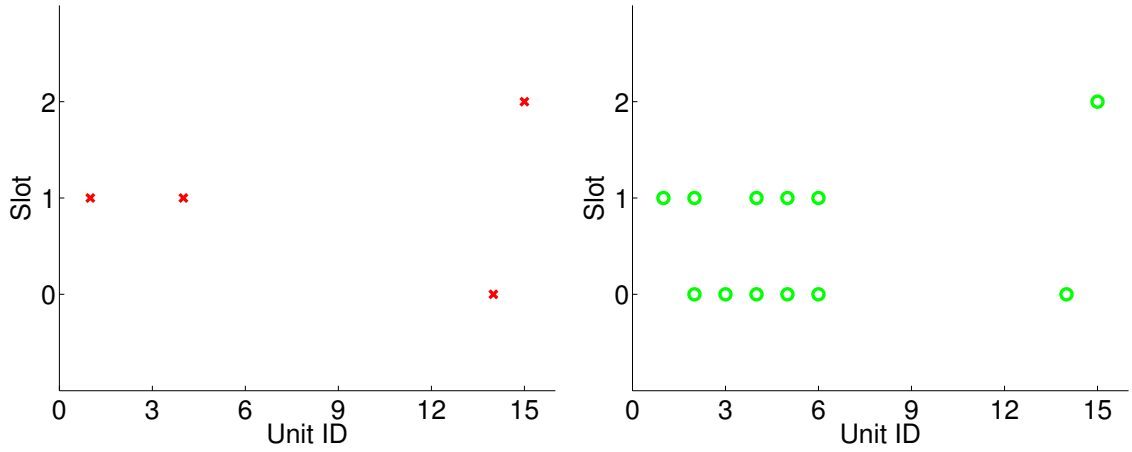
$$\frac{\partial \mu}{\partial \dot{x}} = -\frac{d_T}{\dot{x}^2}(\Pr(H)C_c + \Pr(I)(1-\Pr(H))C_s)l \sum_{\forall\, i,j,q,k} \lambda_i S_{ijqk} y_{jqk} \tag{5.28}$$

Table 5.7 shows the variation in the absolute sensitivity function with respect to speed. Table 5.7 shows that the systems with the linear and nonlinear sensor cost models show the same degree of sensitivity to train speed.

5.3.5 Trade-offs and Sensitivity Analysis with Probability of Event Occurrence

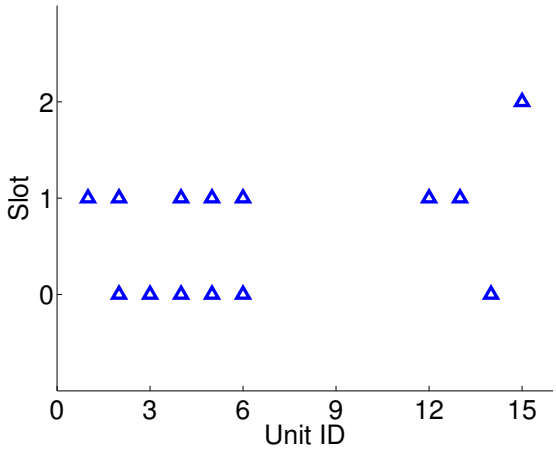
With the next plot we examine the following question: How is the optimal number of sensors for a given train configuration affected by changes in the probability of event occurrence?

Fig. 5.9 shows that the optimal number of sensors needed for a given train configuration depends on the probability of critical event occurrence. When the probability of critical event occurrence is 0.0016, Fig. 5.9a shows that only 4 sensors are needed



(a) Critical Event Probability = 0.0016

(b) Critical Event Probability = 0.0031



(c) Critical Event Probability = 0.0062

Figure 5.9. Optimal Sensor Locations as Critical Event Probability Changes

to achieve the optimal system cost metric. When the probability of critical event occurrence is 0.0031 Fig. 5.9b shows that the optimal number of sensors is 12. Next, Fig. 5.9c shows that the optimal number of sensors is 14 if the probability of critical event occurrence is 0.0062. Doubling the probability of critical event occurrence to 0.0125 means the optimal number of sensors is now 29. Fig. 5.10 shows the relationship between probability of critical event occurrence and the number of sensors. Thus, we conclude that the optimal number of sensors is dependent on the probability

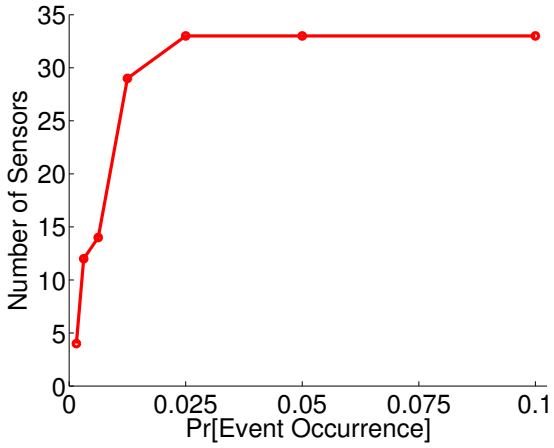


Figure 5.10. Variation in Required Number of Sensors with Probability of Critical Event

of event occurrence, with more sensors required, as expected, when events are more likely.

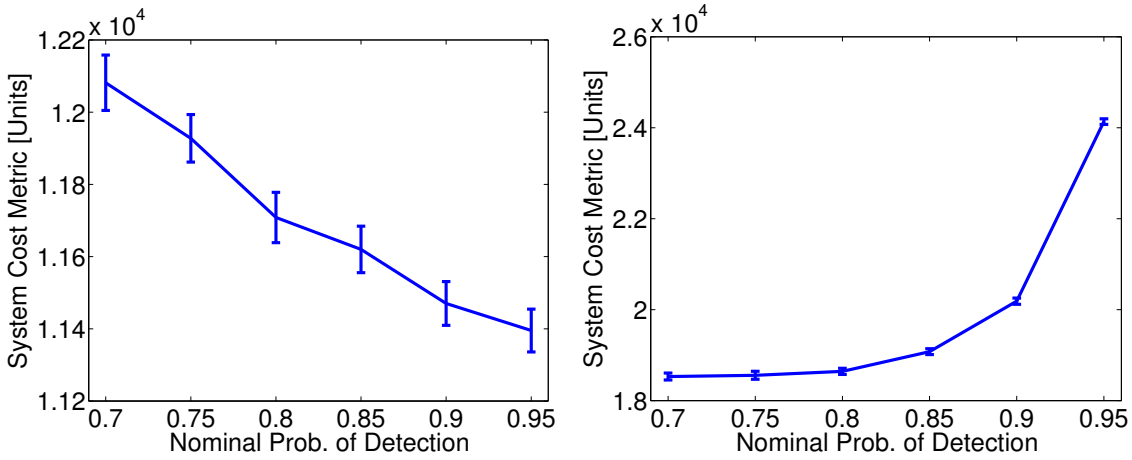
For both the linear and nonlinear sensor cost models the absolute sensitivity function with respect to probability of event occurrence is given by equation (5.29). Equation (5.29) is independent of train speed and it also shows that for a fixed probability of event occurrence, the system cost metric only depends on the train configuration.

$$\frac{\partial \mu}{\partial \zeta} = \left(2 \sum_{\forall j,q,k} \sigma_j y_{jqk} - \sum_{\forall i,j,q,k} (\epsilon + \varphi) \sigma_j S_{ijqk} y_{jqk} \right) \tag{5.29}$$

Further analysis shows that the system cost metric's sensitivity to the probability of event occurrence is independent of train speed, as expected, In addition, for the 33 container train the system cost metric's sensitivity to the probability of event occurrence is fixed at 9.900×10^4 for both the linear and nonlinear sensor cost models.

5.3.6 Effects of Variations in Probability of Detection

Suppose that we lack precise knowledge about the probability of detection offered by the sensors, and all we have is an average and a range. What is the effect of



(a) Linear sensor cost

(b) Nonlinear sensor cost

Figure 5.11. Train-mounted System: Effect of Variations in Sensor Probability of Detection

variations in the probability of detection on the system cost metric? This study can be seen as probabilistic tolerance analysis on the system cost metric, where we "find the distribution of the output from the distribution of the parameters" [91]. In this case we carried out Monte Carlo simulations on a given train configuration. For each nominal probability of detection, e.g., 0.75, 0.80, 0.85, etc., it was assumed that the probability of detection offered by the sensors had a mean value equal to the nominal with a standard deviation of 20%. The simulations were run 101 times for each nominal probability of detection. The mean sensor cost metric and 99% confidence intervals are plotted in Fig. 5.11. From Fig. 5.11 we conclude that there is greater variation in the system cost metric as the nominal (mean) probability of detection gets smaller. Furthermore, the system cost metric is higher for the nonlinear sensor cost model. As a result system designers should bear in mind that as the nominal probability of detection of sensors gets smaller there will be greater variation in the system cost metric.

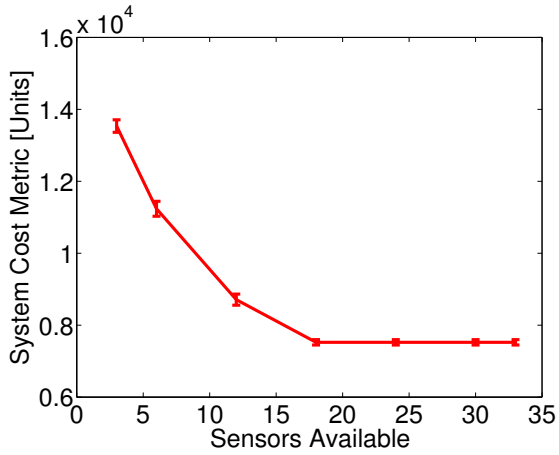
Table 5.8. Container Savings Distributions

Label	Name	Comments
A	High value containers dominate savings	16 low value, 10 medium value, and 7 high value containers.
B	Approximately equal numbers of low & medium value containers	16 low value, 12 medium value, and 5 high value containers.
C	Mostly medium value containers	9 low value, 23 medium value, and 1 high value containers.
D	Mostly low value containers	19 low value, 5 medium value, and 9 high value containers.

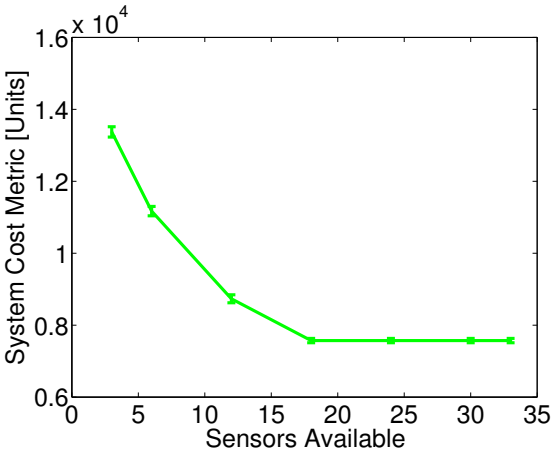
5.3.7 Effects of Different Container Savings Distributions

Finally, we examine the following questions: How does the system cost metric vary for different container savings distributions? Given the same container savings distribution how does the system cost metric vary with changes in the probability of event occurrence? How is the optimal number of sensors assigned affected by changes in the container savings distribution. To answer these questions, suppose we have 33 containers on a train with the distributions shown in Table 5.8. The savings of the low value containers are chosen from a Gaussian distribution with mean 20,000 units and standard deviation 5,000 units. The savings for the medium value containers are chosen from a Gaussian distribution with mean 100,000 units and standard deviation 20,000 units. Finally, the savings for the high value containers are chosen from a Gaussian distribution with mean 200,000 units and standard deviation 50,000 units. Furthermore, suppose that if an event is detected at a container the mean savings in each case listed in Table 5.8 is 80,000 units. Finally, the probability of critical event occurrence at each container is 0.0031. Twenty instances of each container savings distribution shown in Table 5.8 were generated and the system cost metric and sensor assignments computed for each instance.

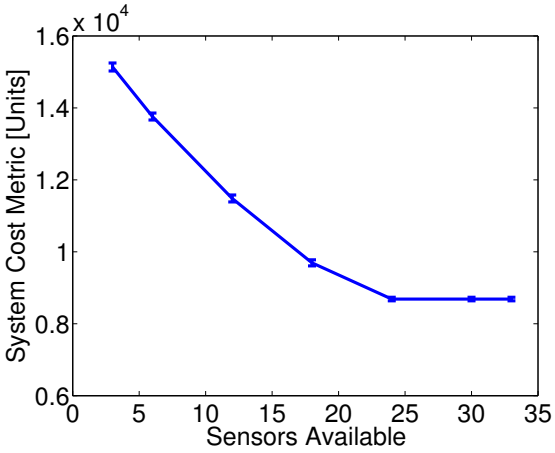
Figs. 5.12 and 5.13 show the average system cost metric for each of these cases,



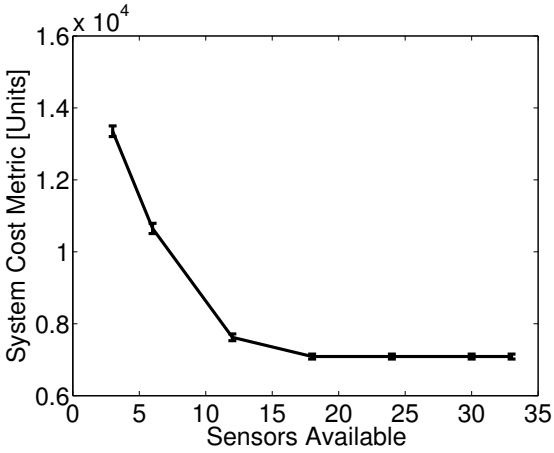
(a) High value containers dominate savings



(b) Approximately equal numbers of low and medium value containers



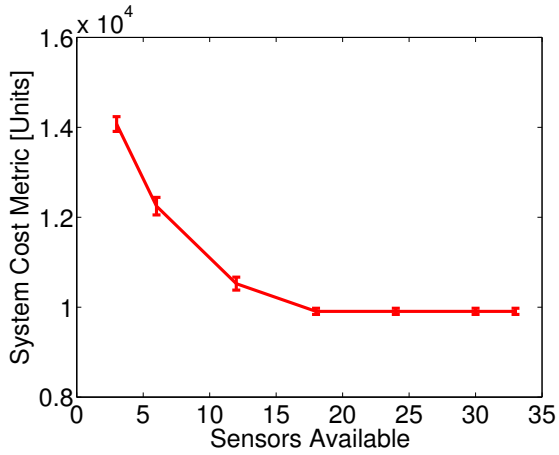
(c) Mostly medium value containers



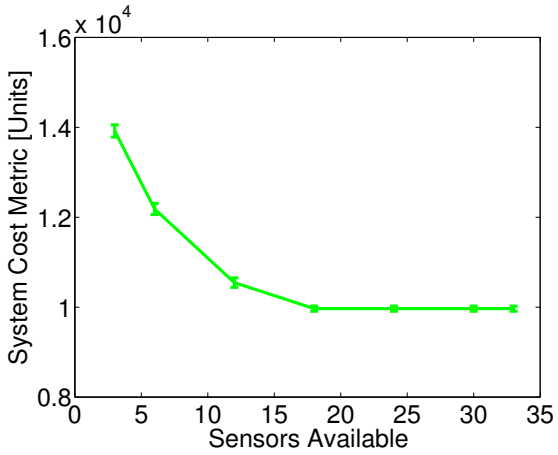
(d) Mostly low value containers

Figure 5.12. Train-mounted System: Different Container Savings Distributions with Linear Sensor Cost Model

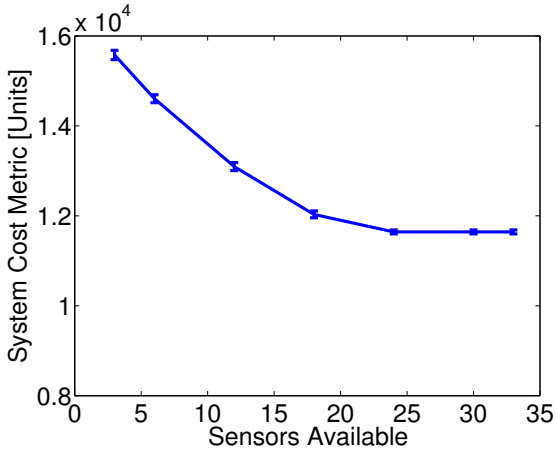
while the 99% confidence interval is represented by the error bars. The confidence intervals become wider as fewer containers have sensors implying that there is more uncertainty in the system as fewer containers have sensors. Figs. 5.12 and 5.13 show that when 24 or more sensors are available, container savings distributions A, B, and D have very similar system cost metrics while container savings distribution C has the highest system cost metric. In all the optimization runs no more than 24 sensors were



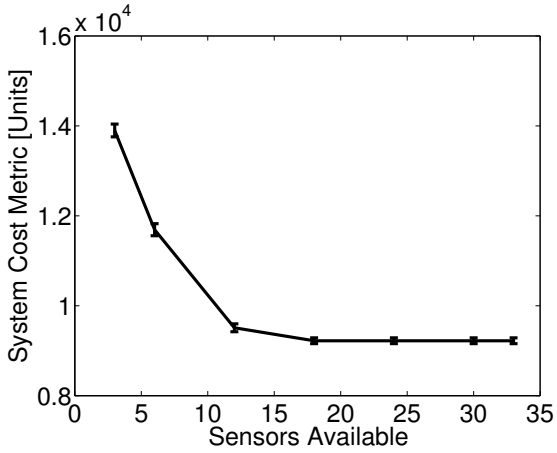
(a) High value containers dominate savings



(b) Approximately equal numbers of low and medium value containers



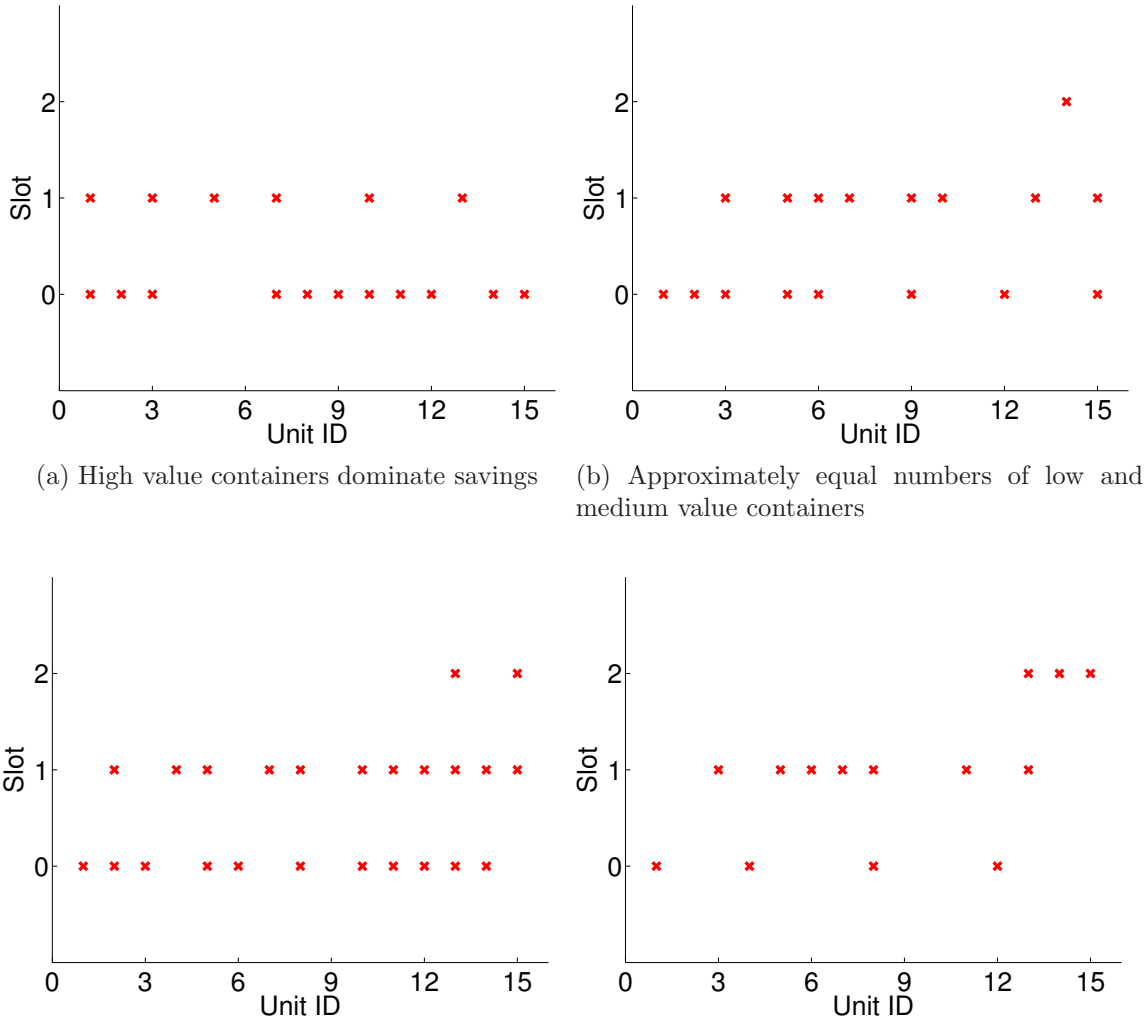
(c) Mostly medium value containers



(d) Mostly low value containers

Figure 5.13. Train-mounted System: Different Container Savings Distributions with Nonlinear Sensor Cost Model

assigned even though there 33 sensors available. Thus, it can be concluded that with a probability of event occurrence of 0.0031 it is not cost-effective to assign sensors to every container. If fewer than 24 sensors are available, then savings distribution C, which has the smallest number of low value containers, has the highest system cost metric. From Figs. 5.12 and 5.13 we conclude that system designers should assign to all the high and medium value containers on the train, as expected. This approach



(a) High value containers dominate savings

(b) Approximately equal numbers of low and medium value containers

(c) Mostly medium value containers

(d) Mostly low value containers

Figure 5.14. Train-mounted System: Optimal Sensor Locations for Different Container Savings Distributions

ultimately leads to a smaller system cost metric with less uncertainty in the final cost metric.

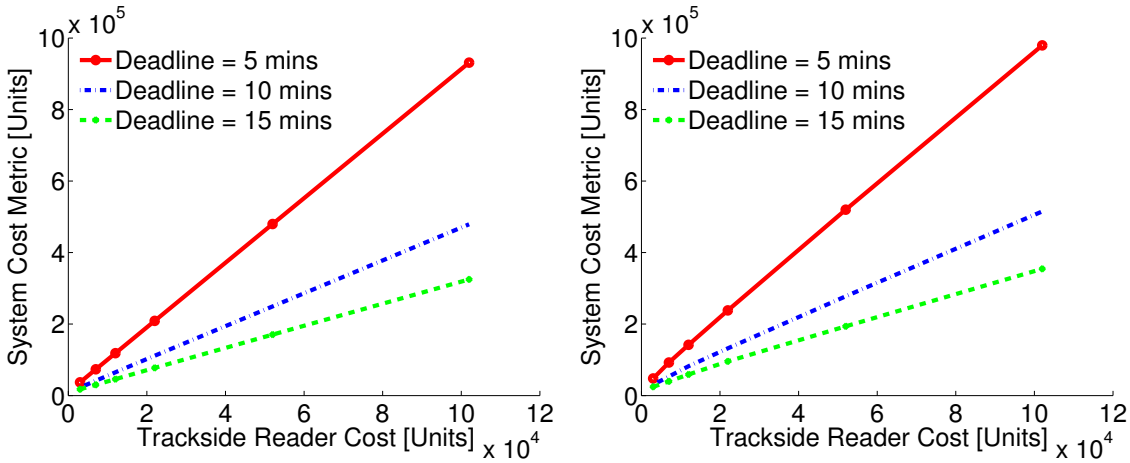
Fig. 5.14 shows the optimal sensor locations for different container savings distributions. For container savings distribution A and B the optimal number of sensors is found to be 17, which is the number of high and medium value containers in each distribution. Figs. 5.14a and 5.14b confirm this observation and show that there are

different sensor mappings for the different savings distributions. For container savings distribution C the optimal number of sensors is found to be 24 as reflected in Fig. 5.14c. Finally, for container savings distribution D the optimal number of sensors is found to be 14. Fig. 5.14d shows how the sensors are mapped for container savings distribution D.

In this section we have presented the trade-offs for the train-mounted deployment of readers and have discussed the parameters and variables have the greatest effect on system cost. We have shown that, as expected, it is important to get sensors with a low probability of false alarm and a high probability of detection. We have also seen that if the specified event notification time is too short, the system cost metric will be high, because there is a low probability that messages can be delivered in a timely fashion. We have also seen that the system cost metric varies very slightly with train speed.

5.4 Trade-offs and Sensitivity Analysis for Trackside System Deployment

In this section we study the system trade-offs and discuss the sensitivity analysis for the trackside deployment of readers. We will also review the system parameters and variables that have the greatest effect on system cost. Unless otherwise stated, the data in this section was generated under the assumptions listed at the start of Section 5.3; however in this case the sensor transmission range is a variable that can be optimized. In addition, there are no repeaters present. Whenever the absolute sensitivity function is computed in this section we will ignore the floor operator in the objective function so that derivatives can be taken.



(a) Linear sensor cost

(b) Nonlinear sensor cost

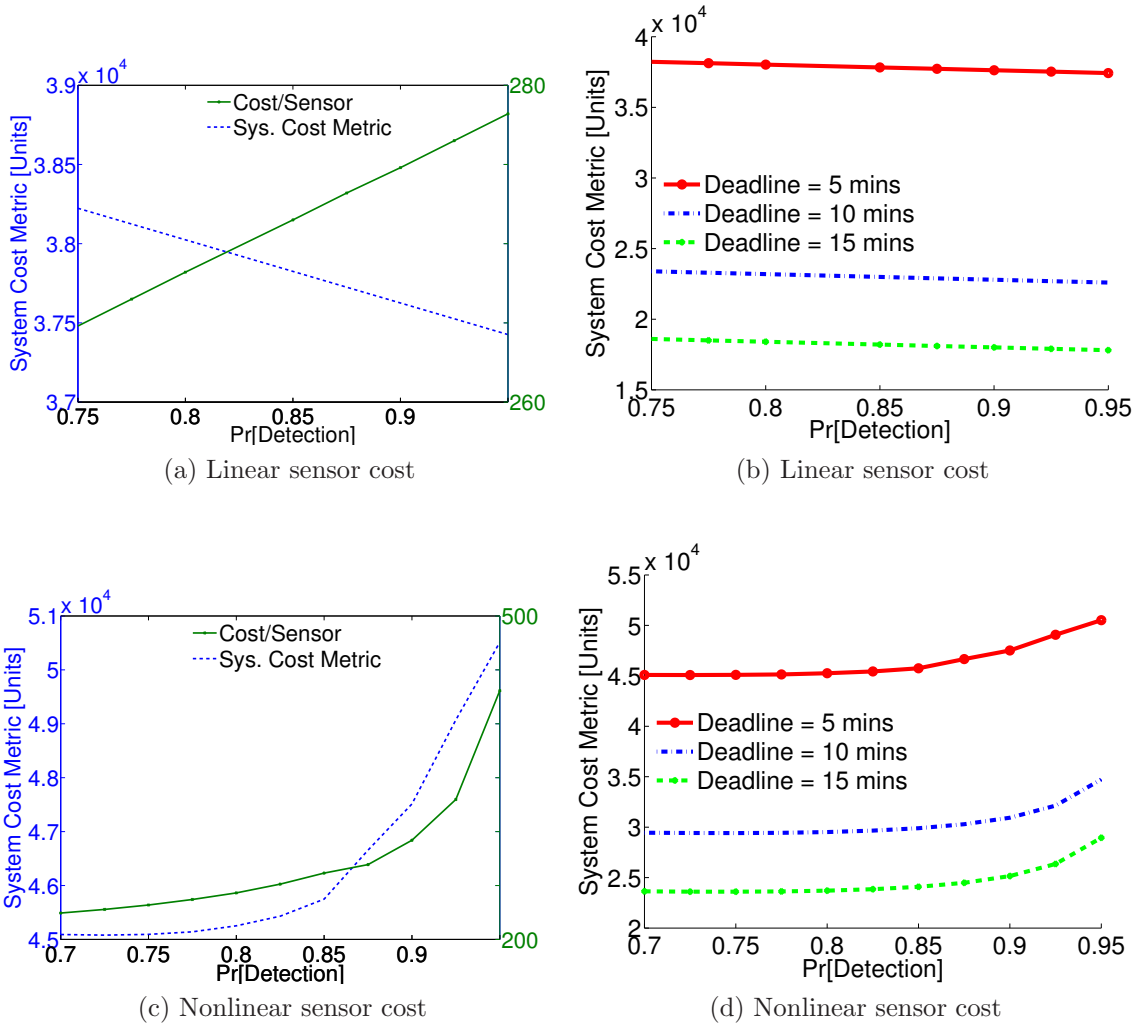
Figure 5.15. Trackside Deployment System: Cost Metric Variation with Reader Cost

5.4.1 Effects of Different Trackside Reader Costs

The first question considered for the trackside system is: What is the effect of changes in the trackside reader cost on the system cost metric? From Fig. 5.15 we see that the cost metric is sensitive to both the trackside reader cost and the reporting deadline. As the reporting deadline grows shorter, the cost metric increases significantly, as expected, as more closely spaced readers are needed to satisfy the event notification deadline.

5.4.2 Trade-offs and Sensitivity Analysis with Probability of Detection

The following questions are examined next: What is the effect of changes in the probability of detection on the system cost metric? How is the system cost affected by changes in the deadline for event notification? For the *linear sensor cost model* Fig. 5.16a shows that the system cost metric decreases, as expected, as the probability of detection offered by the sensors decreases. This decrease in the system cost metric takes place even though the unit cost of each sensor increases with probability of



(a) Linear sensor cost

(b) Linear sensor cost

(c) Nonlinear sensor cost

(d) Nonlinear sensor cost

Figure 5.16. Trackside Deployment System: Cost Metric Variation with Probability of Detection

detection. Fig. 5.16b shows that the system cost metric decreases, as expected, as the deadline for event notification is increased from 5 to 15 minutes. Thus, the system cost metric is sensitive to the event notification time.

For the *nonlinear sensor cost model* Fig. 5.16c shows that the system cost metric decreases as the probability of detection offered by the sensors increases to its optimal value of 0.742, for the probability of event occurrence of 0.0031. Beyond this point the system cost metric increases as the unit sensor cost rises faster. Fig. 5.16d shows

Table 5.9. Trackside Deployment System: Sensitivity Function with respect to Probability of Detection

Event Notification Deadline	Linear	Nonlinear	
	$\frac{\partial \mu}{\partial \epsilon}$	$\frac{\partial \mu}{\partial \epsilon}\Big\vert_{\epsilon=0.80}$	$\frac{\partial \mu}{\partial \epsilon}\Big\vert_{\epsilon=0.85}$
5	-9.679×10^4	-8.868×10^4	-8.066×10^4
10	-9.679×10^4	-8.868×10^4	-8.066×10^4
15	-9.679×10^4	-8.868×10^4	-8.066×10^4

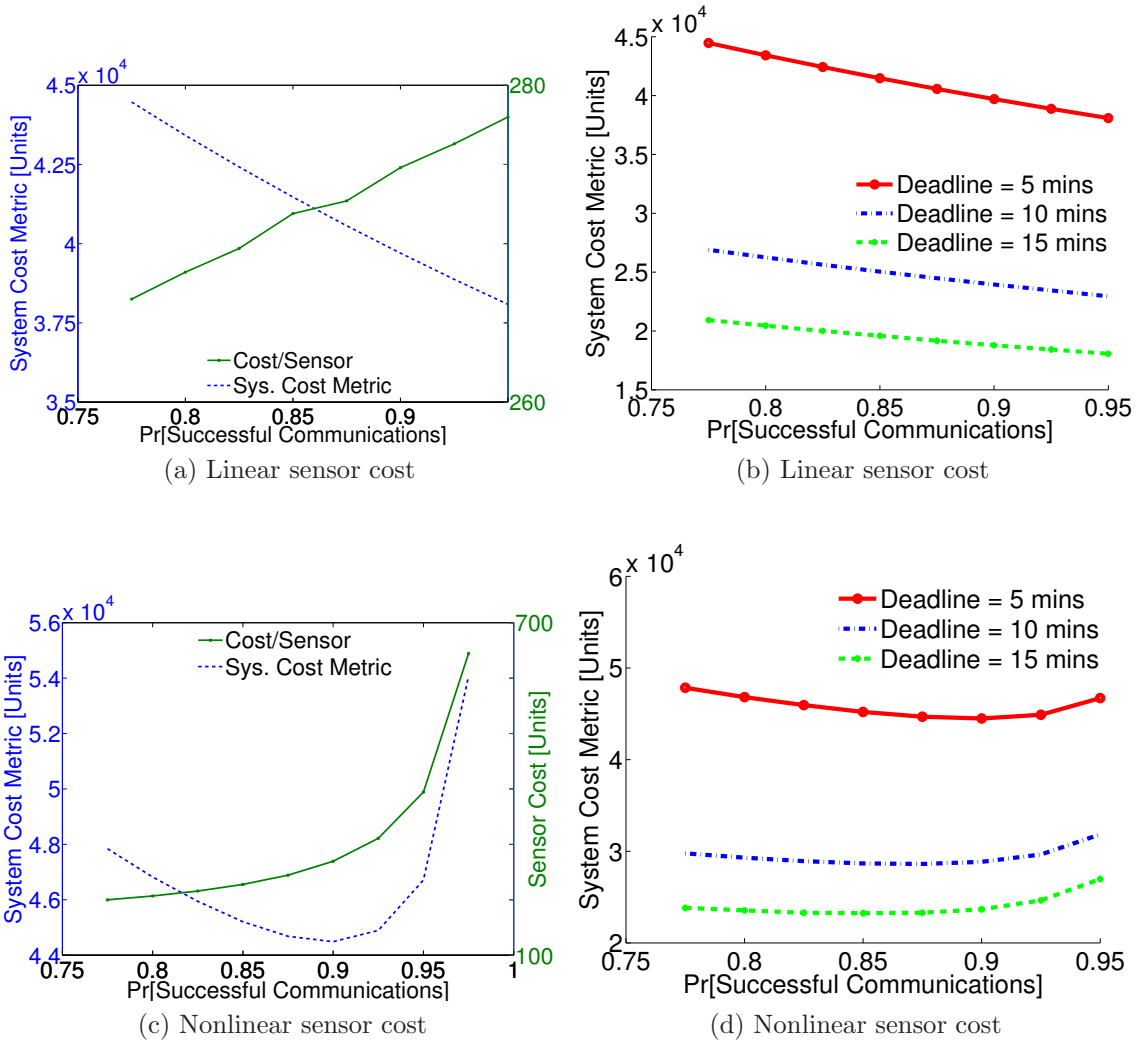
that this trend is also apparent as the event notification time is increased from 5 to 15 minutes.

For the *linear sensor cost model* the absolute sensitivity function with respect to probability of detection is defined as in equation (5.21). Equation (5.21) shows that the system cost metric's sensitivity to the probability of detection is constant for a fixed train configuration and it confirms the finding in Fig. 5.16a. For the *nonlinear sensor cost model* the absolute sensitivity function with respect to the probability of detection is given by equation (5.22). Equation (5.22) shows that the system cost metric has nonlinear sensitivity to the probability of detection, confirming the observations drawn from Fig. 5.16c.

Table 5.9 shows the variation in the absolute sensitivity function with respect to probability of detection as the critical event notification deadline is changed. Table 5.9 shows, as expected, that the event notification deadline has no effect on the absolute sensitivity function with respect to probability of detection.

5.4.3 Trade-offs and Sensitivity Analysis with Probability of Successful Communications

Next, we consider the following questions: What is the effect of changes in the probability of successful communications on the system cost metric? How is the system cost affected by changes in the deadline for event notification?



(a) Linear sensor cost

(b) Linear sensor cost

(c) Nonlinear sensor cost

(d) Nonlinear sensor cost

Figure 5.17. Trackside Deployment System: Cost Metric Variation with Probability of Successful Communications

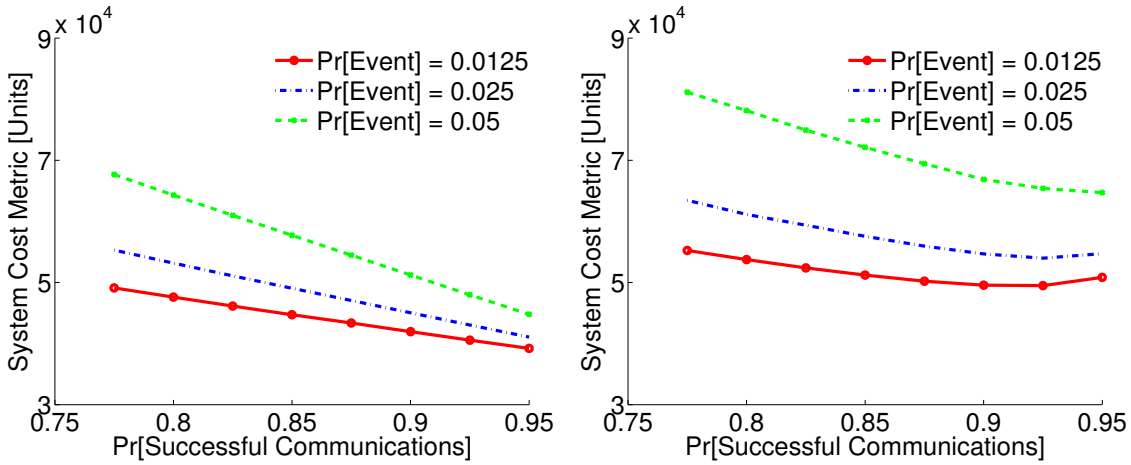
From Figs. 5.17a and 5.17c we see that the system cost metric is sensitive to the probability of successful communications between a sensor and a trackside reader as well as the event notification deadline. For the linear sensor cost model as the probability of successful communications increases, there is greater likelihood that seal events will be reported in a timely manner, and hence the decline in the cost metric. For the nonlinear sensor cost model the system cost metric decreases as the probability of successful communications from a sensor to a trackside reader rises to

0.93. After this point the system cost metric increases.

For Figs. 5.17a and 5.17c the optimization models were run for each value of the probability of successful communications and the results were compiled. The results show that when the probability of successful communications is low the optimum system cost metric is achieved by selecting sensors with high transmission ranges so that the readers can be placed far apart. It is these variations in the sensor transmission range and probability of successful communications that lead to the shape of the cost per sensor line in Fig. 5.17a. The same trade-off between sensor transmission range and all of the other sensor characteristics is also taking place in Fig. 5.17c. Again, we see that when the probability of successful communications is low the sensor transmission ranges are high; however, the cost per sensor rises as the probability of successful communications increases due to the nonlinear sensor cost model. In this case the optimal probability of successful communications is 0.935, which explains the system cost metric curve getting flatter beyond this point.

Fig. 5.18 shows the effect of probability of successful communications on the system cost metric as the probability of critical event occurrence is changed while the critical event notification deadline is fixed at 5 minutes. Fig. 5.18a shows that the system cost metric decreases, as expected, as the probability of critical event occurrence is reduced. Fig. 5.18b shows that for the nonlinear sensor cost model and a fixed event notification deadline, as the probability of successful communications is increased the system cost metric drops until the probability of successful communications achieves the optimal value for the given probability of event occurrence. In Fig. 5.18b the optimal probability of successful communications is 0.871 and 0.935 when the probability of event occurrence is 0.0125 and 0.05, respectively.

Equation (5.30) shows the absolute sensitivity function with respect to probability



(a) Linear Sensor Cost

(b) Nonlinear Sensor Cost

Figure 5.18. Trackside Deployment System: Cost Metric variation with Prob. of Successful Communications

of successful communications for the linear sensor cost model. Note that the plot of the system cost metric against probability of successful communications in Figs. 5.17a and 5.17b is nonlinear and gradually becomes flatter as the probability of successful communications increases. Equation (5.30) shows that for a fixed train configuration the cost metric's sensitivity to the probability of successful communications depends on the event notification time. This result confirms an observation from Fig. 5.17b, where we observed that the slope of the system cost metric with respect to probability of successful communications decreased with an increase in event notification time.

$$\frac{\partial \mu}{\partial \rho} = \sum_{\forall i,j,q,k} \left(\mathrm{FP}_3 - \zeta \sigma_j \right) S_{ijqk} y_{jqk} - \left(\frac{C_A + C_{\mathrm{AD}}}{t_f \times \mathrm{LT}_A} + \frac{C_{\mathrm{BC}} + C_{\mathrm{BD}}}{t_f \times \mathrm{LT}_c} \right) \frac{d_T \dot{x} (\tau - \sum_{r=2}^{5} \tilde{t}_r)}{\left(\dot{x} \rho \left(\tau - \sum_{r=2}^{5} \tilde{t}_r \right) + 2\theta \right)^2} \quad (5.30)$$

For the nonlinear sensor cost model equation (5.31) shows the absolute sensitivity function with respect to probability of successful communications. This equation is

Table 5.10. Trackside Deployment System: Sensitivity Function with respect to Probability of Successful Communications

Event Notification Deadline	Linear		Nonlinear	
	$\frac{\partial \mu}{\partial \rho}\Big\vert_{\rho=0.80}$	$\frac{\partial \mu}{\partial \rho}\Big\vert_{\rho=0.85}$	$\frac{\partial \mu}{\partial \rho}\Big\vert_{\rho=0.80}$	$\frac{\partial \mu}{\partial \rho}\Big\vert_{\rho=0.85}$
5	-1.338×10^5	-1.299×10^5	-1.292×10^5	-1.168×10^5
10	-1.182×10^5	-1.157×10^5	-1.103×10^5	-9.983×10^4
15	-1.111×10^5	-1.094×10^5	-1.031×10^5	-9.341×10^4

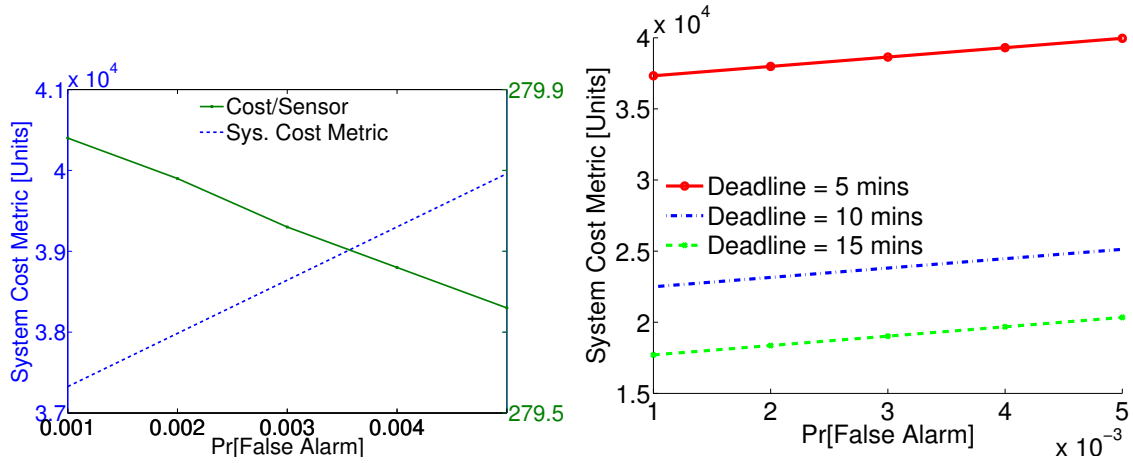
nonlinear confirming the observation from Figs. 5.17c and 5.17d that the slope of the system cost metric varies nonlinearly. Furthermore, the sensitivity varies with the train configuration and the weight of the sensor cost allocated to improving the probability of successful communications.

$$\frac{\partial \mu}{\partial \rho} = \sum_{\forall i,j,q,k} \left(\frac{\mathrm{FP}_3}{(1-\rho)^2} - \zeta \sigma_j \right) S_{ijqk} y_{jqk} - \left(\frac{C_A + C_{\mathrm{AD}}}{t_f \times \mathrm{LT}_A} + \frac{C_{\mathrm{BC}} + C_{\mathrm{BD}}}{t_f \times \mathrm{LT}_c} \right) \frac{d_T \dot{x} (\tau - \sum_{r=2}^{5} \tilde{t}_r)}{\left(\dot{x} \rho \left(\tau - \sum_{r=2}^{5} \tilde{t}_r \right) + 2\theta \right)^2} \quad (5.31)$$

Table 5.10 shows the values of the absolute system cost metric sensitivity function with respect to probability of successful communications at selected points. From Table 5.10 it can be concluded that the system cost metric is more sensitive to the probability of successful communications under the linear sensor cost model.

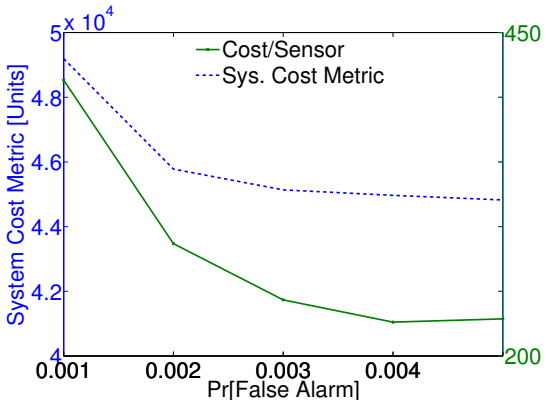
5.4.4 Trade-offs and Sensitivity Analysis with Probability of False Alarm

We examine the following questions next: What is the effect of changes in the probability of false alarm on the system cost metric? How is the system cost metric affected by changes in the deadline for event notification? Fig. 5.19a shows that as the probability of false alarm offered by each sensor rises, the system cost metric increases, as expected. This increase in the system cost metric takes place even though the unit sensor cost declines marginally. Fig. 5.19b shows that the system cost metric changes

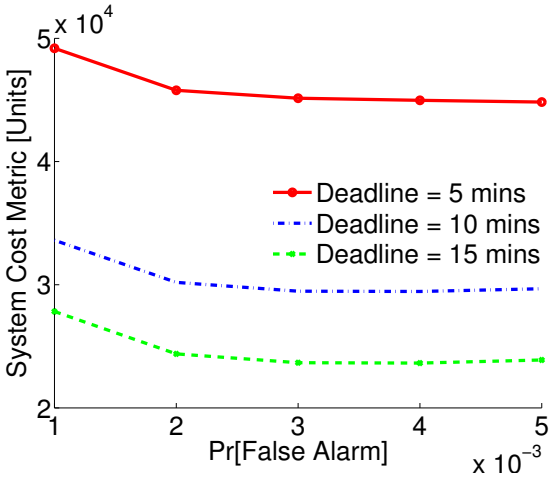


(a) Linear sensor cost

(b) Linear sensor cost



(c) Nonlinear sensor cost



(d) Nonlinear sensor cost

Figure 5.19. Trackside Deployment System: Cost Metric Variation with Probability of False Alarm

with the event reporting deadline, just as we saw with the probability of detection. For the nonlinear sensor cost model Fig. 5.19c shows that the system cost metric decreases as the probability of false alarm offered by each sensor drops to about 0.0035. As the probability of false alarm rises beyond this point the more frequent false alarms cause an increase in the system cost metric. Fig. 5.19d shows that as the event notification deadline is increased the system cost metric becomes smaller,

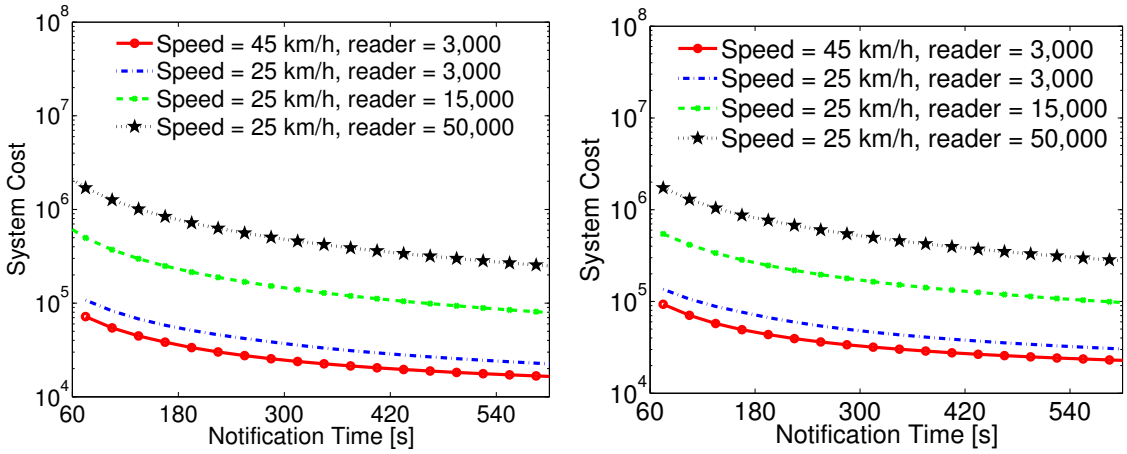
as expected. In addition, for each event notification deadline the optimal probability of false alarm is found to be 0.0035.

For the *linear sensor cost model* the absolute sensitivity function with respect to probability of false alarm is defined as in equation (5.26). Equation (5.26) shows that the system cost metric's sensitivity to the probability of false alarm is constant for a fixed train configuration confirming the observation from Fig. 5.19a that the slope of the system cost metric plot is constant with respect to probability of false alarm. For the *nonlinear sensor cost model* the absolute sensitivity function with respect to the probability of false alarm is given by equation (5.27). Equation (5.27) shows that the system cost metric has nonlinear sensitivity to the probability of false alarm, confirming the observations drawn from Fig. 5.19c. Comparing Figs. 5.17a and 5.19a we conclude that the trackside system is more sensitive to the probability of false alarm than to the probability of successful communications. Thus, we conclude that the emphasis should be placed on purchasing sensors with as low a probability of false alarm, followed by a high probability of successful communications, and then as high a probability of detection as possible.

Table 5.11 shows the variation in the absolute system cost metric sensitivity function with respect to probability of false alarm. As was seen for the train-mounted system deployment, Table 5.11 shows that the system with a nonlinear sensor cost model shows less sensitivity to the probability of false alarm if the selected sensors have a probability of false alarm close to the optimal value of 0.0035.

Table 5.11. Trackside Deployment System: Sensitivity Function with respect to Probability of False Alarm

Event Notification Deadline	Linear	Nonlinear	
	$\frac{\partial \mu}{\partial \alpha}$	$\frac{\partial \mu}{\partial \alpha}\Big\|_{\alpha=0.002}$	$\frac{\partial \mu}{\partial \alpha}\Big\|_{\alpha=0.004}$
5	6.583×10^5	-1.403×10^6	1.444×10^5
10	6.583×10^5	-1.403×10^6	1.444×10^5
15	6.583×10^5	-1.403×10^6	1.444×10^5



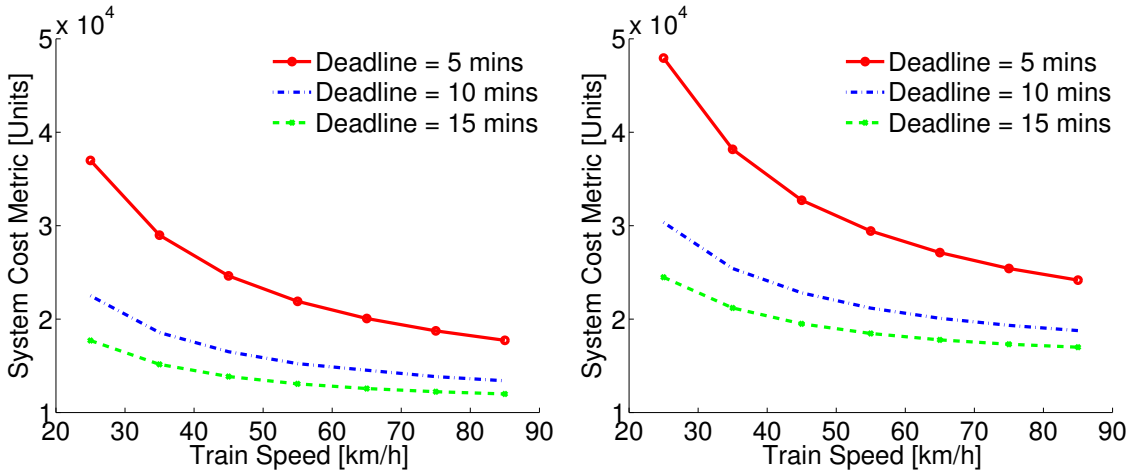
(a) Linear sensor cost

(b) Nonlinear sensor cost

Figure 5.20. Trackside Deployment System: Cost Metric Variation with Event Notification Time

5.4.5 Trade-offs between System Cost Metric and Desired Critical Event Notification Time

It is important to understand the relationship between the desired critical event notification time and the system cost metric. Fig. 5.20 allows us to answer the following question: How is the system cost metric affected by changes in the event notification time? Figs. 5.20a and 5.20b show that the system cost metric decreases, as expected, as the event notification deadline is increased. As the event notification deadline is increased the trackside readers can be placed further apart leading to the decline in the system cost metric. Fig. 5.20 also shows that for a fixed trackside reader cost the system cost metric is lower, as expected, if the train travels at a higher



(a) Linear sensor cost

(b) Nonlinear sensor cost

Figure 5.21. Trackside Deployment System: Cost Metric Variation with Train Speed

speed. This is because for the same event notification deadline trackside readers can be placed further apart at higher train speeds.

5.4.6 Trade-offs and Sensitivity Analysis with Train Speed

We now consider the following question: What are the effects of changes in train speed on the system cost metric? Fig. 5.21 shows how the system cost metric varies with speed for the linear and nonlinear sensor cost models when the event notification deadline is 5 minutes and each trackside reader costs 3000 units. System optimization shows that for a notification time of 5 minutes the readers will be spaced every 1.94 km for a train speed of 25 km/h and every 6.53 km for a train speed of 85 km/h. Fig. 5.21 shows that the system with the nonlinear sensor cost model will be more expensive to deploy than one with a linear sensor cost model. Secondly, we note that there is a nonlinear relationship between train speed and the system cost metric. Equation (5.32) shows the absolute sensitivity function with respect to train speed for the linear sensor cost model. This function is clearly not linear and it confirms

Table 5.12. Trackside Deployment System: Sensitivity Function with respect to Train Speed

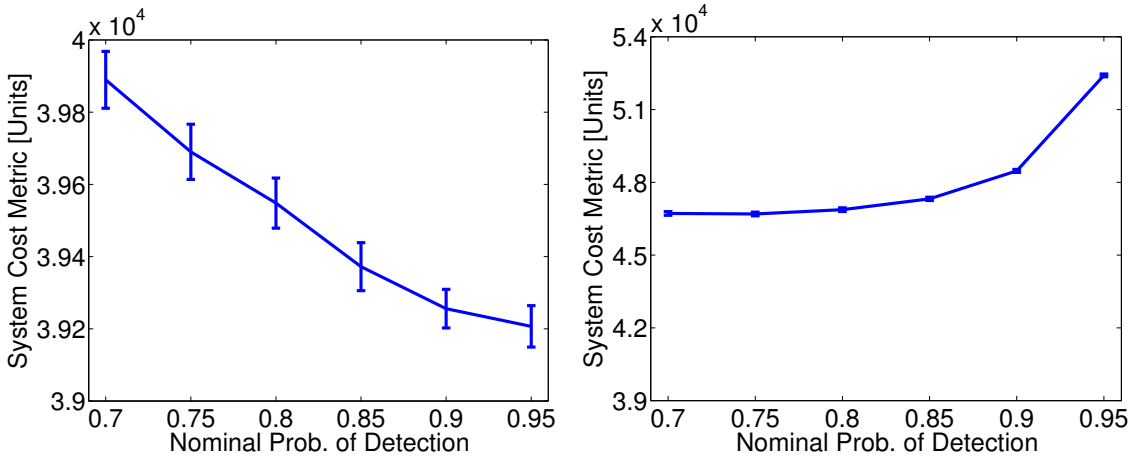
Event Notification Deadline	Linear		Nonlinear	
	$\frac{\partial \mu}{\partial \dot{x}}\Big\|_{\dot{x}=35}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\|_{\dot{x}=75}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\|_{\dot{x}=35}$	$\frac{\partial \mu}{\partial \dot{x}}\Big\|_{\dot{x}=75}$
5	-5.148×10^2	-9.494×10^1	-5.773×10^2	-1.279×10^2
10	-2.252×10^2	-3.071×10^1	-2.990×10^2	-6.337×10^1
15	-1.191×10^2	-9.651×10^0	-1.932×10^2	-4.211×10^1

that a nonlinear relationship exists between the system cost metric and train speed. The absolute sensitivity function with respect to train speed for the nonlinear sensor cost model is shown in equation (5.33). Equation (5.33) is nonlinear confirming the previous result that the relationship between train speed and system cost metric is not linear.

$$\frac{\partial \mu}{\partial \dot{x}} = \sum_{\forall i,j,q,k} \left(\zeta \beta \sigma_j - \frac{d_T}{\dot{x}^2} C_c l \lambda_i - \beta \mathrm{FP}_3 \right) S_{ijqk} y_{jqk} + \left(\frac{C_A + C_{\mathrm{AD}}}{t_f \times \mathrm{LT}_A} + \frac{C_{\mathrm{BC}} + C_{\mathrm{BD}}}{t_f \times \mathrm{LT}_c} \right) \times \frac{d_T (\dot{x}\beta - \rho)(\tau - \sum_{r=2}^{5} \tilde{t}_r)}{\left(\dot{x}\rho \left(\tau - \sum_{r=2}^{5} \tilde{t}_r \right) + 2\theta \right)^2} \quad (5.32)$$

$$\frac{\partial \mu}{\partial \dot{x}} = \sum_{\forall i,j,q,k} \left(\zeta \beta \sigma_j - \frac{d_T}{\dot{x}^2} C_c l \lambda_i - \frac{\beta \mathrm{FP}_3}{(1-\rho)^2} \right) S_{ijqk} y_{jqk} + \left(\frac{C_A + C_{\mathrm{AD}}}{t_f \times \mathrm{LT}_A} + \frac{C_{\mathrm{BC}} + C_{\mathrm{BD}}}{t_f \times \mathrm{LT}_c} \right) \times \frac{d_T (\dot{x}\beta - \rho)(\tau - \sum_{r=2}^{5} \tilde{t}_r)}{\left(\dot{x}\rho \left(\tau - \sum_{r=2}^{5} \tilde{t}_r \right) + 2\theta \right)^2} \quad (5.33)$$

Table 5.12 shows the values of the absolute sensitivity function with respect to the train speed at selected points for the linear and nonlinear sensor cost models. For both models the system cost metric becomes less sensitive to speed as train speed increases. In addition, Table 5.12 shows that the system with the nonlinear sensor cost model is marginally more sensitive to train speed.



(a) Linear sensor cost

(b) Nonlinear sensor cost

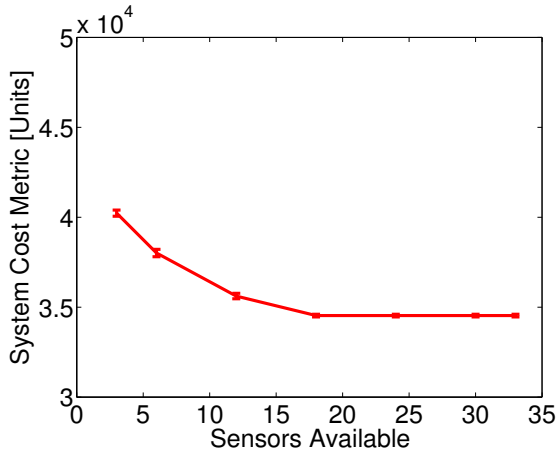
Figure 5.22. Trackside Deployment System: Effect of Variation in Probability of Detection

5.4.7 Effects of Variations in Probability of Detection

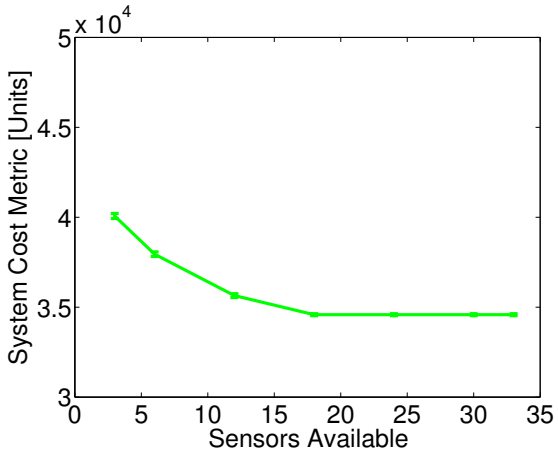
Next, we consider the following question: Suppose that we lack precise knowledge about the probability of detection offered by the sensors, and all that we have is an average and a range. What is the effect of these variations on the system cost metric? As was done in Section 5.3.6 we carry out Monte Carlo simulations to determine the effect of these variations. Fig. 5.22 shows the mean system cost metric and 99% confidence intervals for the trackside model. In this case there is a standard deviation of 20% in the minimum specification for probability of detection. As expected, the results here are similar to Fig. 5.11, where we observed that there was greater variation in the system cost metric as the nominal probability of detection is decreased.

5.4.8 Effects of Different Container Savings Distributions

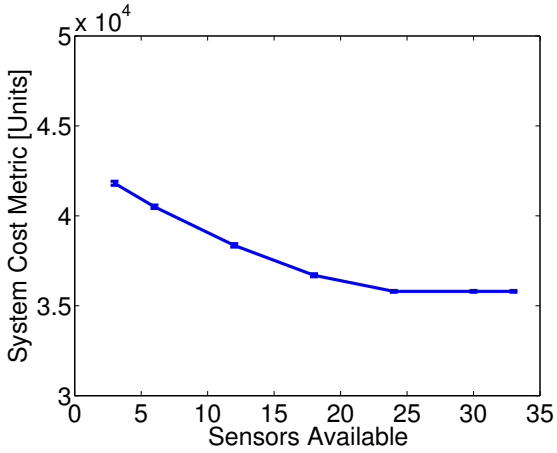
Finally, we examine the following questions: How does the system cost metric vary for different container savings distributions? Given the same container savings



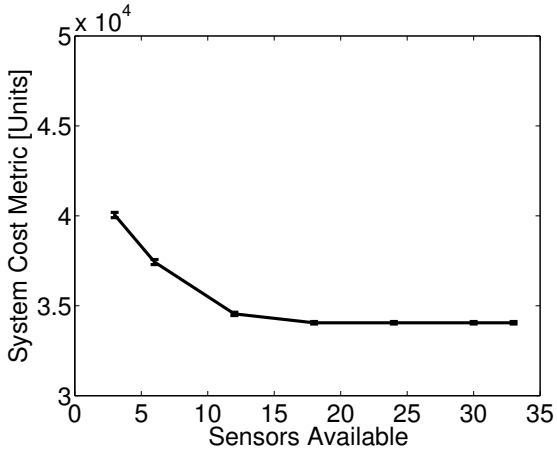
(a) High value containers dominate savings



(b) Approximately equal numbers of low and medium value containers



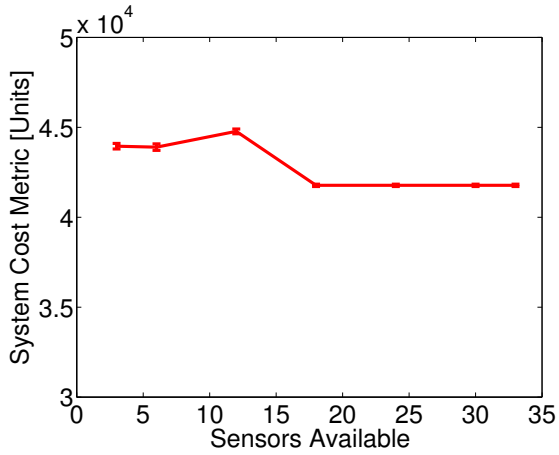
(c) Mostly medium value containers



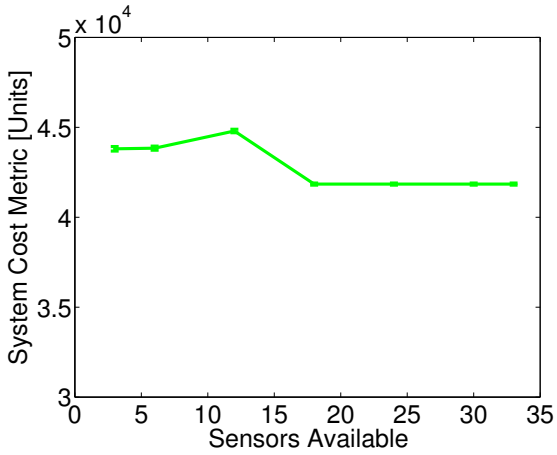
(d) Mostly low value containers

Figure 5.23. Trackside Deployment System: Different Container Savings Distributions with Linear Sensor Cost Model

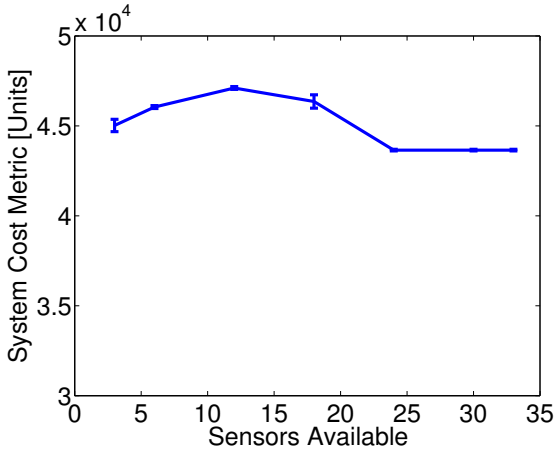
distribution how does the system cost metric vary with changes in the deadline for event notification? Suppose we have 33 containers on a train with the same distributions and means shown in Section 5.3.7. As was done in Section 5.3.7 we generated 20 instances of each container distribution and calculated the mean and variance of the system cost metric. Figs. 5.23 and 5.24 show the means and 99% confidence intervals. These plots show that, just as for the train-mounted system, the system cost metric



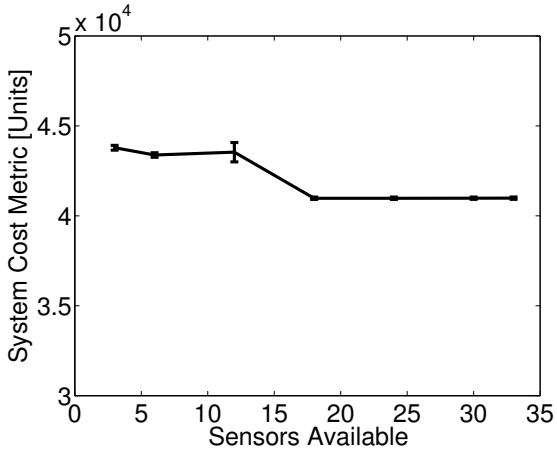
(a) High value containers dominate savings



(b) Approximately equal numbers of low and medium value containers



(c) Mostly medium value containers



(d) Mostly low value containers

Figure 5.24. Trackside Deployment System: Different Container Savings Distributions with Nonlinear Sensor Cost Model

for container savings distributions A, B, and D is very similar as long as 24 or more sensors are available to be placed on the containers. Container savings distribution C, which has the highest number of medium value containers, has the highest system cost metric always. When the nonlinear sensor cost model is used Fig. 5.24 shows that the system cost metric has its maximum value when about 12 containers have sensors. This surprising result comes about as follows: when only 3 to 12 sensors

are available, the system cost metric is minimized by selecting a sensor transmission range that is as high as possible so that the trackside readers can be placed far apart. When 3 to 12 containers get sensors, it is these expensive sensors getting used. As a result the system cost metric increases with the number of sensors. When more than 12 sensors are available, then cheaper sensors can be selected resulting in the shapes of the graphs shown in Fig. 5.24. Figs. 5.23 and 5.24 also show that the confidence intervals get wider as fewer containers have sensors. The rest of our observations are identical to those drawn in Section 5.3.

In this section we have reviewed the trade-offs for the trackside deployment of readers to monitor cargo on a train. We have seen that the system is most sensitive to the probability of false alarm. Thus, system designers ought to choose sensors with the lowest probability of false alarm, followed by a high probability of successful communications, and a high probability of detection. We have also seen that the system cost metric increases very quickly with trackside reader cost. Finally, we observed that the system cost metric is inversely related to the train speed, since trackside readers can be placed further apart as train speed increases.

5.5 Trade-offs Involving Train-Mounted and Trackside System Deployments

In this section we compare the train-mounted and trackside deployments and discuss some of the trade-offs between these alternative system designs. In order to compare the systems fairly the following conditions are assumed for both system deployments: the train speed is 25 km/h, a commercial-off-the-shelf sensor in each case costs 250 units, and the probability of critical event occurrence at each container is 0.0031. In the train mounted case we assume that the sensor transmission range

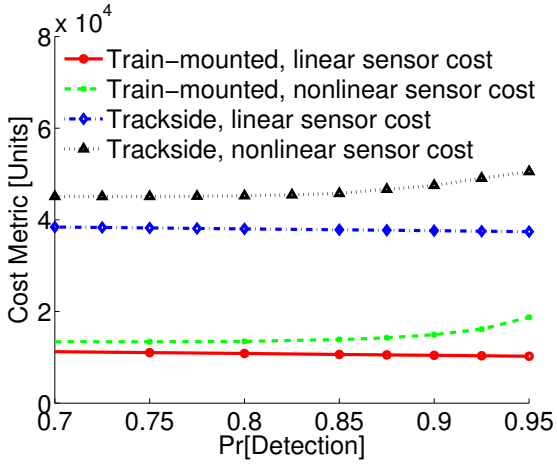
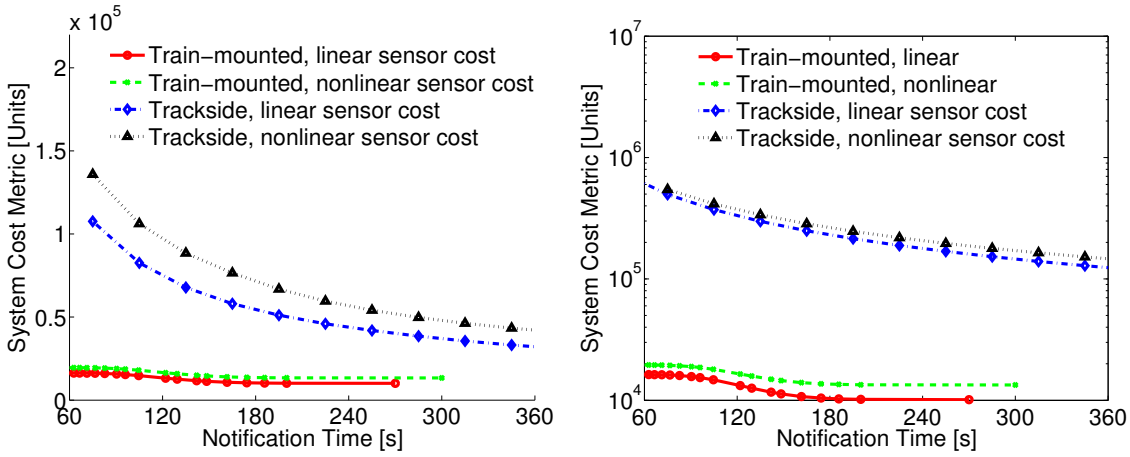


Figure 5.25. Comparison of Train-mounted and Trackside Deployment Systems: Probability of Detection

is 50 m and that there are repeaters present on every third car. In the trackside case we assume that the sensor transmission range is a variable that can be optimized. As we have done before we will compare the two systems as the sensors' probability of detection changes, as the decision maker notification time is varied, as the sensors' probability of false alarm changes, as the probability of critical event occurrence changes, as the train speed changes, and as the savings distribution of the containers changes.

5.5.1 Comparing Train-Mounted and Trackside System Deployments as Probability of Detection Changes

The first question we examine is: When comparing the train-mounted and trackside reader deployment systems how is the system cost metric affected by changes in the probability of detection? Fig. 5.25 compares the cost metric for the train-mounted and trackside deployment systems as the probability of detection offered by the sensors changes. Fig. 5.25 shows that the trackside system has a higher cost metric due to the higher expenses associated with deploying several trackside readers.



(a) Reader Cost = 3,000 units

(b) Reader Cost = 15,000 units

Figure 5.26. Comparison of Train-mounted and Trackside Deployment Systems: Notification Time

5.5.2 Comparing Train-Mounted and Trackside System Deployments as Decision Maker Notification Time Changes

Secondly, we consider the following question: When comparing the train-mounted and trackside reader deployment systems, how is the system cost metric affected by changes in the maximum time required to notify decision makers of events? Note that for the trackside system deployment notification times of 15–240 s are too short to be practical, but these times are included in the discussion for illustrative purposes. Similarly for the train-mounted system the modeling work from Chapter 3 shows that there is a 47% chance of timely notification in less than 120 s; however, these times are also included for the purposes of discussion. Fig. 5.26a compares the train-mounted and trackside deployment systems with regards to the maximum specified notification time when each trackside reader costs 3,000 units, while Fig. 5.26b makes the comparison when each reader costs 15,000 units. Fig. 5.26a shows that there is no event notification time such that the trackside reader deployment system can be favored over the train-mounted system. Fig. 5.26b shows, as expected, that the

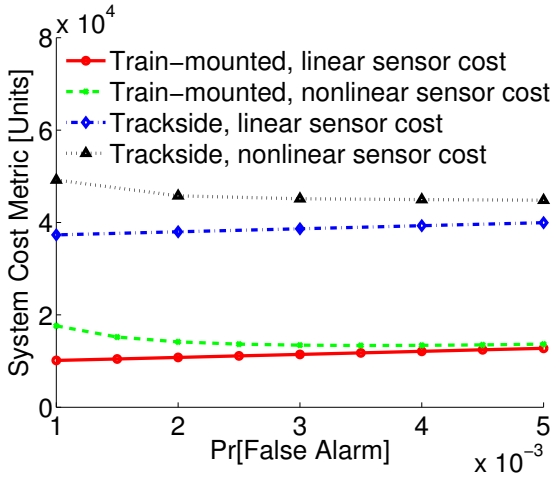


Figure 5.27. Comparison of Train-mounted and Trackside Deployment Systems: Probability of False Alarm

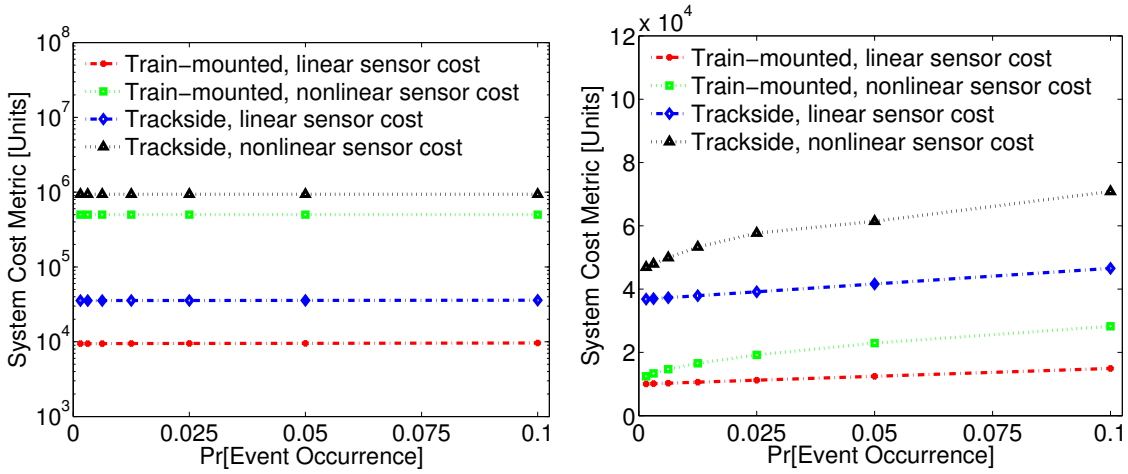
difference between the cost metrics for both systems becomes more acute as the reader cost increases.

5.5.3 Comparing Train-Mounted and Trackside System Deployments as Probability of False Alarm Changes

Next, we study the following question: When comparing the train-mounted and trackside deployment systems how is the cost metric affected by changes in the probability of false alarm? Fig. 5.27 shows that regardless of the sensor cost model that is applied, the trackside system has a higher cost metric than the train-mounted system when compared with respect to probability of false alarm.

5.5.4 Comparing Train-Mounted and Trackside System Deployments as Probability of Event Occurrence Changes

Before deploying a cargo monitoring system it is important to understand how the system will behave as the probability of event occurrence changes. As a result we study the following question: when comparing the train-mounted and trackside



(a) Almost perfect sensors

(b) Imperfect sensors

Figure 5.28. Comparison of Train-mounted and Trackside Deployment Systems: Probability of Event Occurrence for Almost Perfect and Imperfect Sensors

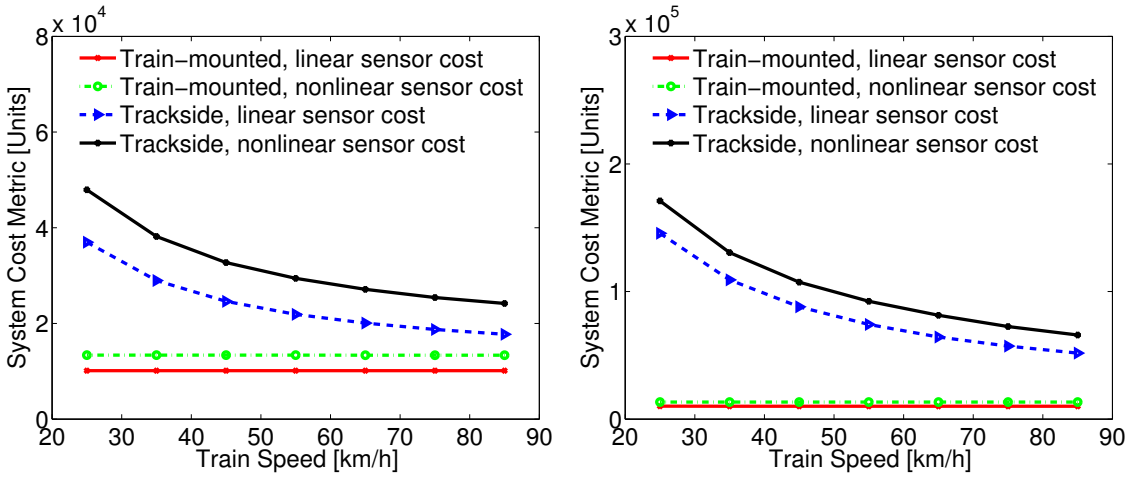
systems how is the system cost affected by changes in the probability of critical event occurrence? Fig. 5.28 shows how the system cost metric varies with probability of event occurrence for the trackside and train-mounted systems. The almost perfect sensors used in this case have probabilities of detection and false alarm of 0.999 and 0.0001, respectively while the imperfect sensors have probabilities of detection and false alarm of 0.975 and 0.001, respectively. Fig. 5.28a shows that when almost perfect sensors are used, the trackside system with nonlinear sensor cost is more expensive to deploy than the trackside system with linear sensor cost. In addition, the train-mounted system with nonlinear sensor cost is more expensive to deploy than the trackside system with linear sensor cost. Finally, from Fig. 5.28a we conclude that if we have almost perfect sensors, the system cost metric remains almost constant regardless of the probability of event occurrence. The observant reader might wonder why the system cost metric does not vary much with probability of event occurrence. When almost perfect sensors are used the cost of a missed detection is very small

due to the sensors' very high probability of detection. As a result, for both the train-mounted and trackside deployment systems the cost metric is dominated by the costs of the network elements, i.e., the sensors and backhaul device in the case of the train-mounted system and the sensors and the cost of the readers in the trackside case.

From Fig. 5.28b shows that if the sensors are imperfect, then both instances of the trackside system have a higher cost metric than the train-mounted system. Furthermore, the system cost metric increases as events become more likely on a trip. For the train-mounted system with the linear sensor cost model there is a 48% increase in the system cost metric as the probability of critical event occurrence at a container increases from 0.0015 to 0.1. For the trackside system with the nonlinear sensor cost model the corresponding increase is 50.1% when the event notification deadline is 5 minutes.

5.5.5 Comparing Train-Mounted and Trackside System Deployments as Train Speed Changes

Average train speed and trackside reader cost are all important factors in determining the system cost metric. As a result when comparing the train-mounted and trackside deployment systems we need to understand how the system cost metric is affected by changes in train speed. Fig. 5.29 shows how the cost metric varies with train speed for the train-mounted and trackside deployment systems. Fig. 5.29a compares the cost metrics for the trackside and train-mounted models. For the trackside system the reader cost is 3,000 units and this instance is compared with the train-mounted system when the train has a 90% probability of being in satellite or cellular coverage. From Fig. 5.29a we observe that since only the messaging costs vary with



(a) Reader Cost = 3,000 units

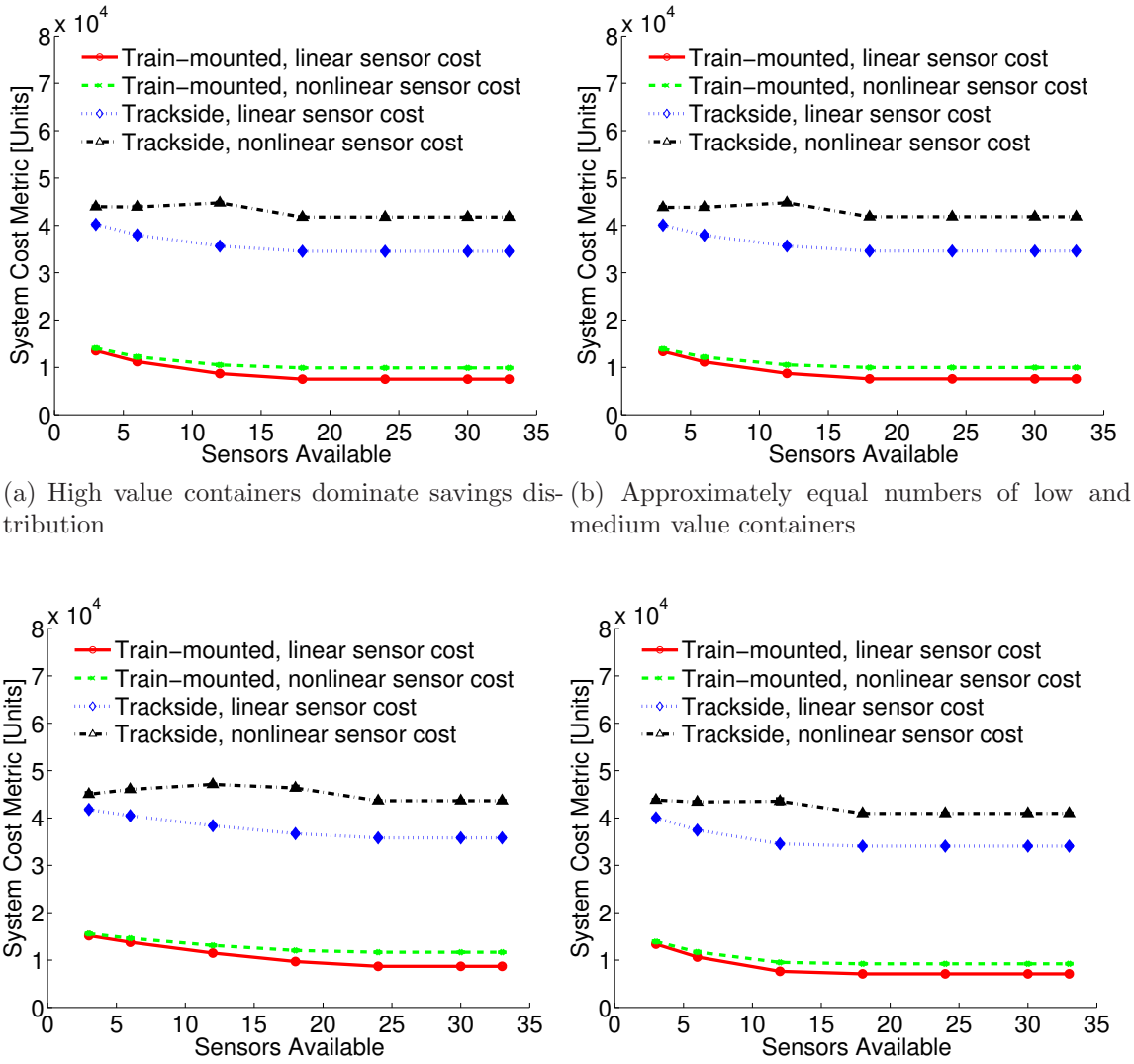
(b) Reader Cost = 15,000 units

Figure 5.29. Comparison of Train-mounted and Trackside Deployment Systems: Effect of Train Speed and Mode of Communications

trip duration for the train-mounted deployment this system is less sensitive to train speed, whereas with the trackside system the messaging costs, reader separation, and consequently the number of readers are affected by train speed. As a result the trackside system is much more sensitive to train speed. Fig. 5.29b compares the cost metric for a train-mounted system where the train has a 90% probability of being in satellite or cellular coverage to instances of the trackside system where the trackside readers cost 15,000 units. As was previously observed, the trackside system is sensitive to train speed. From Fig. 5.29 we conclude that there is no train speed between 25 and 90 km/h such that the trackside deployment system has a lower cost metric than the train mounted system.

5.5.6 Comparing Train-Mounted and Trackside System Deployments as Container Savings Distributions Change

Finally, we need to understand how the system costs vary for different container savings distributions when comparing the train-mounted and trackside deployment



(a) High value containers dominate savings distribution

(b) Approximately equal numbers of low and medium value containers

(c) Mostly medium value containers

(d) Mostly low value containers

Figure 5.30. Comparison of Train-mounted and Trackside Deployment Systems: Different Container Savings Distributions

systems. Fig. 5.30 aids us in answering this question, where the container savings distributions are defined as in Table 5.8. In Fig. 5.30 the 99% confidence intervals are represented by the widths of the lines in the plots. Fig. 5.30 shows that when the containers are drawn from the same savings distribution the cost metric for the train-mounted and trackside systems exhibit the same general trends with the trackside

system always costing more than the train-mounted system. Furthermore, the difference in cost metrics for trackside and train-mounted systems implemented on the same set of containers is approximately constant. This difference is mainly composed of the amortized cost of the trackside readers.

In this section we have compared the train-mounted and trackside deployment systems with respect to probabilities of detection and false alarm, event notification time, probability of event occurrence and train speed. We have found that the trackside system has a higher cost metric than the train-mounted system in all the cases we considered. The results also show that the trackside deployment case is much more sensitive to train speed than the train-mounted deployment. Finally, the results show that if the train-mounted and trackside systems are deployed to containers drawn from the same container savings distribution, the trackside system deployment costs more than the train-mounted system deployment. In addition, the difference in cost metrics for the trackside and train-mounted systems is approximately constant.

5.6 Trade-offs for Train-Mounted System Deployment with Periodic Callback

It is also possible to have a deployment case where the sensors and readers are on the train, but unlike in Section 5.3 the backhaul communications device "calls" home periodically. Such a deployment scenario was used in the long-haul trial [85]. Our models cannot currently address this deployment scenario, and we defer the extension of the models to tackle this deployment scenario to future work. However, we make some observations on potential advantages of this deployment scenario and its trade-offs. The train-mounted deployment of readers with periodic callback has an advantage over the trackside case in that the hardware investment needed to deploy

the periodic callback system is much less than that needed for the trackside case. There will be a trade-off between the frequency of callbacks and the time window within which events must be reported. The shorter the time window when events need to be reported, the higher the frequency of callbacks. It should be noted that even if the frequency of callbacks is high, if the train is moving through an area with poor Internet connectivity, then messages can be delayed for extended periods awaiting transmission. For example, Kuehnhausen and Frost [85] observed that some messages spent up to 1273 s in the message queue awaiting transmission because a connection could not be established to the virtual network operations center. It is possible to reduce some of these queueing delays by determining the satellite and cellular coverage along a given train route. This information can then be used to determine an appropriate callback frequency for timely reporting of events. The telecommunications coverage information can also be used to determine the best locations for calling the operations center.

5.7 Conclusion

In this chapter we have studied the trade-offs that exist when monitoring cargo in motion. Only a limited portion of the design space for implementing a cargo monitoring system has been explored, due to the size of the design space. However, the trade-offs study shows the power of the models developed in Chapter 4, furthermore, the study has shown those parameters and variables that should be of greatest concern to the system designer. The strength of the models has also been underscored by the fact that we were able to use two different sensor cost models for the train-mounted and trackside deployment systems and have arrived at optimal system parameters in each case. Our studies in this chapter have shown that:

- The system deployment cost is inversely related to the deadline for decision maker notification. The system deployment cost is also inversely related to the average train speed, with the trackside model being most greatly affected by the train speed. Thus, system designers deciding to implement either the train-mounted or trackside systems can trade-off train speed and event notification deadline with system cost, though in general the train companies would always go as fast as possible.

- For the train-mounted system emphasis should be paid to the probability of false alarm before the probability of detection. For the trackside system more emphasis should be paid to the probability of false alarm, then the probability of successful communications, and then the probability of detection.

- The optimal number of sensors is dependent on the container savings distribution and the probability of critical event occurrence. However, in a real system deployment would most likely have to use real and "dummy" sensors on all the containers to prevent cargo thieves from knowing which containers to target.

- With the *nonlinear sensor cost* model, the optimal probability of detection is dependent on the probability of critical event occurrence. For the *linear sensor cost* model, on the other hand, the optimal probability of detection is independent of the probability of critical event occurrence.

- The optimal probability of false alarm is independent of the probability of critical event occurrence and event reporting deadline.

There are also other trade-offs, that are not easily quantifiable, which have not been covered in this chapter. For example, a trackside deployment of network elements allows several trains to be monitored using the same readers and backhaul

components. However, network elements deployed, particularly in remote locations, are subject to being stolen or vandalized. Furthermore, trains cannot be monitored on parts of the rail network that are not equipped with readers and backhaul communications network elements. A train-mounted deployment of network elements, on the other hand, allows for continuous monitoring of cargo subject to the availability of communications signals.

In this chapter we studied the system trade-offs for a relatively small train with 33 containers and 15 units. Lai *et al.* [19] indicate that some intermodal trains have up to 115 units and 244 containers. As a result, Chapter 6 presents a heuristic for determining appropriate sensor characteristics and assigning sensors to containers on a train.

Chapter 6

A Heuristic for Design of Communications Systems and Networks for Monitoring Cargo in Motion along Trusted Corridors

6.1 Introduction and Motivation

Mixed Integer Nonlinear Programs (MINLP) can be used to optimally assign sensors to containers on trains and perform system trade-off studies as seen in Chapters 4 and 5. Unfortunately, mixed integer nonlinear programs are $\mathcal{NP}$-hard [18] and as we saw in Chapter 4 we were only able to find optimal sensor locations for relatively small[1] trains with 15 units and 33 containers. However, Lai *et al.* [19] state that an international intermodal stack train can have up to 104 units and 224 containers. Thus, a method needs to be devised to choose the best (or close to best) way to deploy sensors to realistic trains. The purpose of this chapter is to answer the following questions: Can a heuristic be created to assign sensors to containers on typical intermodal trains? How does the heuristic's performance compare to the optimization approach?

The rest of this chapter is laid out as follows: in Section 6.2 a heuristic for placing sensors on containers for cargo monitoring is presented. Section 6.3 validates the heuristic using a representative train. Section 6.3 also shows the results from the

[1]Note that the train size was limited by the computer used to solve the optimization problem.

Table 6.1. Symbols Used in Heuristic

Symbol	Comments
σ_j	Savings observed at container j if an event is detected and reported in a timely manner.
F_j	Visibility requirement for probability of false alarm at container j.
E_j	Visibility requirement for probability of detection at container j.
TR_j	Visibility requirement for probability of timely reporting at container j.
θ	Sensor transmission range.
α	Sensor's probability of false alarm.
ϵ	Probability of detection offered by sensor.
θ	Sensor transmission range.
ζ	Probability of a critical event at a container.
w_j	Visibility weight of container j.
C_α	Cost of one false alarm.
$\mathrm{FA}_L, \mathrm{FA}_U$	Sensor manufacturer's lower and upper limits on probability of false alarm.
$\mathrm{PD}_L, \mathrm{PD}_U$	Sensor manufacturer's lower and upper limits on probability of detection.
C_H	Acquisition cost of one sensor.
FP_1	Weight of sensor cost allocated to improving event detection.
FP_2	Weight of sensor cost allocated to reducing false alarms.
FP_4	Weight of sensor cost allocated to improving sensor transmission range.
C_F	Fixed portion of sensor cost.

heuristic's application to a typical intermodal train. Finally, concluding remarks are provided in Section 6.4.

6.2 Heuristic Description

This section describes a heuristic for assigning sensors to containers on an intermodal train. First, the symbols used in the heuristic are defined in Table 6.1. Next, the assumptions underlying the heuristic are presented. Finally, the heuristic is presented.

In order for the heuristic to execute successfully, the following conditions are assumed to be true:

- There is a finite number of sensors available. The number of sensors available does not exceed the number of containers.

- An sensor placement solution exists for the given set of containers and sensors.
- The unit cost of each sensor is related to the sensor capabilities using either a linear or nonlinear cost model.
- The transmission range of the sensors can be modified so that all the sensors are connected in a cargo monitoring network.
- There is a visibility weight associated with each container. The visibility weight combines the savings resulting from event detection and the container's visibility requirements into a single metric. This metric can then be used in a ranking scheme to determine which container gets a sensor next. The visibility weight, w_j is defined as the product of each container's savings, σ_j, and the length of the vector, $(1 - F_j, E_j, \mathrm{TR}_j)$, with each container's visibility requirements.

$$w_j = \sigma_j \sqrt{(1 - F_j)^2 + E_j^2 + \mathrm{TR}_j^2} \tag{6.1}$$

Using these assumptions the heuristic shown in Fig. 6.1 can be developed.

Fig. 6.1 shows the algorithm for assigning sensors to containers on a train. First, the algorithm stores the number of sensors available to be used on the train and the total savings for all the containers on the train. The visibility weight for each container is computed and then sorted in descending order. Next, the probabilities of detection and false alarm for each sensor are computed. For the linear sensor cost model, Chapter 5 showed that it is best to have the probability of detection as high as possible while the probability of false alarm should be as low as possible. For the nonlinear sensor cost model the best probability of detection can be selected when the absolute sensitivity function with respect to the probability of detection is zero, i.e., when equation (5.22) is zero. Similarly, the best probability of false alarm

$i_{\max} \leftarrow$ Number of sensors
TotSavings $\leftarrow \sum_j \sigma_j$
for all containers **do**
  $w_j \leftarrow \sigma_j \sqrt{(1-F_j)^2 + E_j^2 + \mathrm{TR}_j^2}$
  {Compute a visibility weight metric that is the product of the savings resulting from detecting an event and the square-root of the sum of the square of the visibility components.}
end for
Sort containers by descending order of visibility weight metric
{Compute the initial sensor characteristics.}
if Using Linear Sensor Cost model **then**
  $\alpha \leftarrow \mathrm{FA}_L$
  $\epsilon \leftarrow \mathrm{PD}_U$
else
  $\alpha \leftarrow \sqrt{\dfrac{\mathrm{FP}_2}{C_\alpha}}$
  $\epsilon \leftarrow 1 - \sqrt{\dfrac{i_{\max} \times \mathrm{FP}_1}{\mathrm{TotSavings} \times \zeta}}$
end if
{Assign sensors to containers.}
$i \leftarrow 1$
for all containers **do**
  SavingsAtRisk $\sigma_j \zeta$
  if $w_j \neq 0$ AND $i < i_{\max}$ AND SavingsAtRisk $> C_H$ **then**
    $S_{ijqk} \leftarrow 1$ {Where $j, q,$ and k come from the container location mapping}
    $i \leftarrow i + 1$
  end if
end for
for all sensors **do**
  Check that each sensor can communicate with its neighbors.
end for
Compute system cost metric for resulting sensor assignment using equation (4.14).

Figure 6.1. Algorithm for Sensor Assignment

occurs when the absolute sensitivity function with respect to the probability of false alarm is zero, i.e., when equation (5.27) is zero. Solving these equations yields the desired probabilities of detection and false alarm. Next, sensors are assigned to the containers in descending order of visibility weight. Note that sensors are assigned as long as there are sensors available to assign and the sensor costs less than the savings that would be lost in an event that is not detected. Next the heuristic checks that each sensor can communicate with its neighbors. A sensor is defined as being within

range of its neighbors if the distance to either of its neighbors is less than the sensor's transmission range. Once this process is complete, the system cost metric for the sensor assignment is computed and the algorithm terminates.

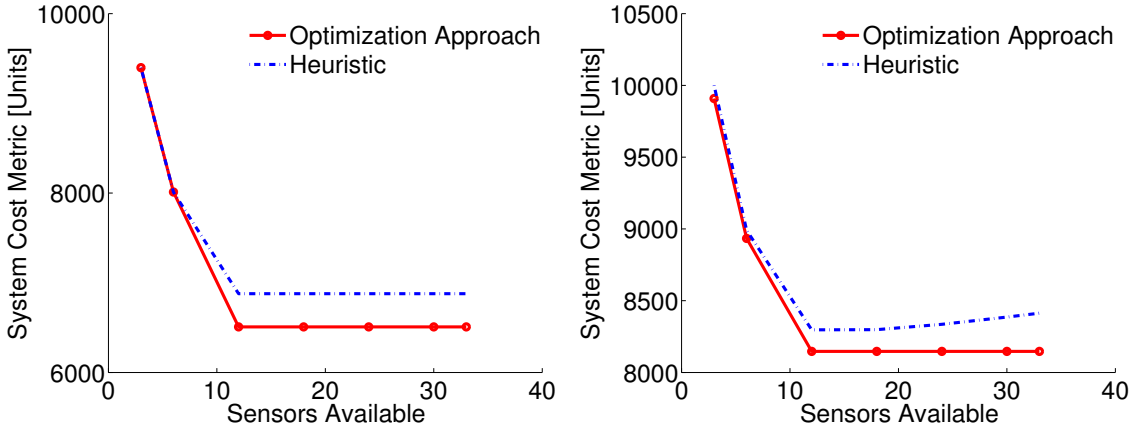
6.3 Heuristic Validation and Application

This section discusses how the heuristic was validated and then presents the results showing the heuristic's application on a typical intermodal train.

6.3.1 Heuristic Validation

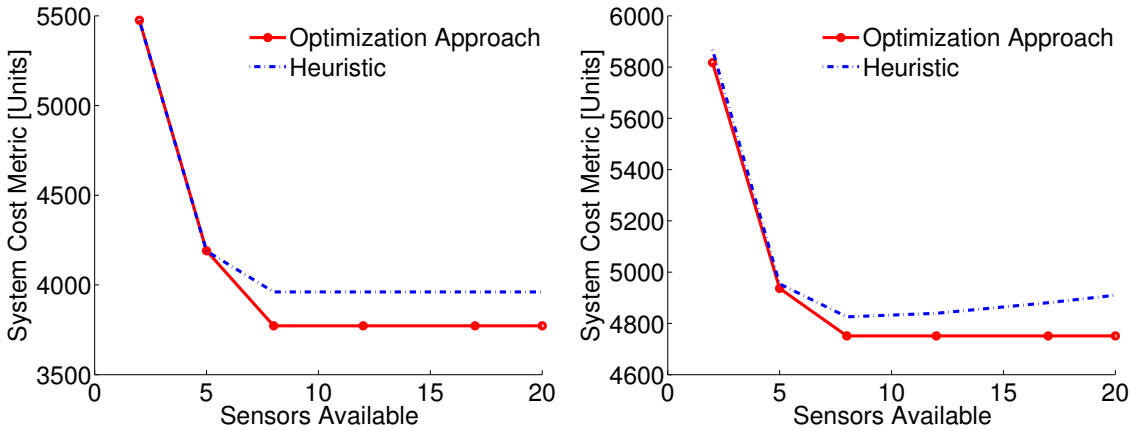
This subsection discusses the validation of the heuristic. The heuristic described in Sec 6.2 was implemented in Java and run over trains of different sizes with the linear and nonlinear sensor cost models. The validation was done using the following assumptions:

- The average train speed was 25 km/h, which is computed from [86, 87].
- The length of the rail trip was 1984 km, which is the distance from Laredo to Kansas City [87].
- The trains were of the following sizes and composition:
  - The first train had 33 containers and 15 units. In this case 21 containers had a mean value of 20,000 units, 9 containers had a mean value of 100,000 units, and 3 containers had a mean value of 200,000 units.
  - The next train had 20 containers and 9 units. In this case 13 containers had a mean value of 20,000 units, 5 containers had a mean value of 100,000 units, and 2 containers had a mean value of 200,000 units.



(a) Linear Sensor Cost (b) Nonlinear Sensor Cost

Figure 6.2. Validation of the Sensor Assignment Heuristic for Train with 33 containers

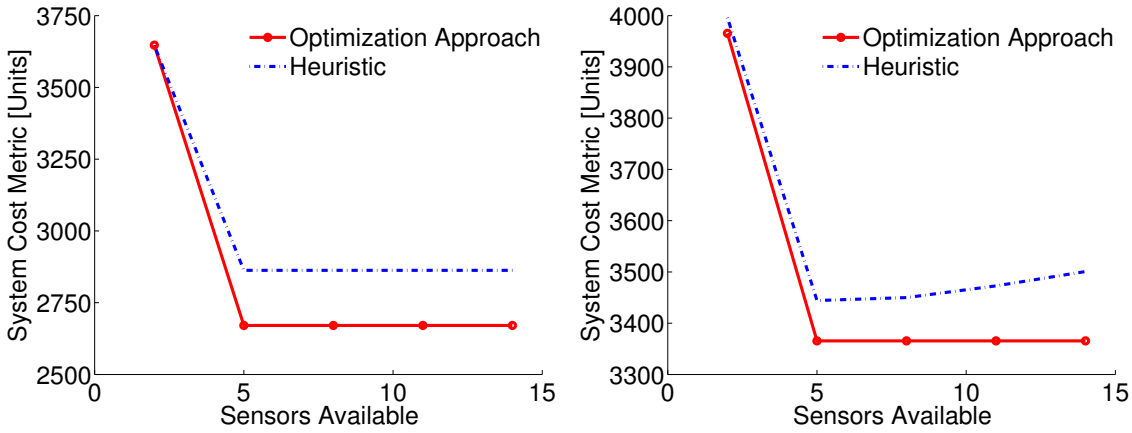


(a) Linear Sensor Cost (b) Nonlinear Sensor Cost

Figure 6.3. Validation of the Sensor Assignment Heuristic for Train with 20 containers

  - The last train had 14 containers and 6 units. In this case 9 containers had a mean value of 20,000 units, 4 containers had a mean value of 100,000 units, and 1 container with a value of 200,000 units.

- The probability of a critical event, such as a container seal being opened, closed, or tampered with, occurring at each container was 3.125×10^{-3}.



(a) Linear Sensor Cost

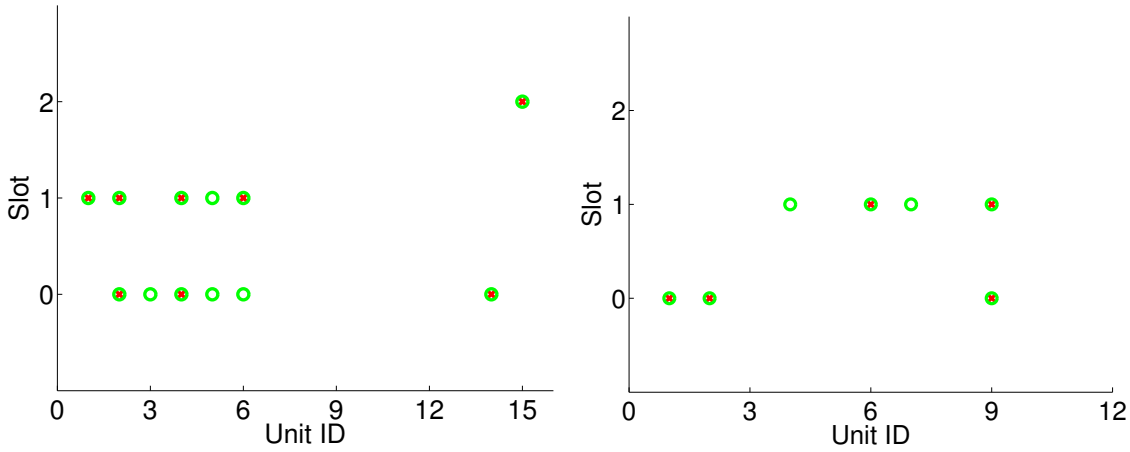
(b) Nonlinear Sensor Cost

Figure 6.4. Validation of the Sensor Assignment Heuristic for Train with 14 containers

For the train with 33 containers, Fig. 6.2a shows that under the linear sensor cost model the optimization and heuristic approaches agree if 3 or 6 sensors are used. When more sensors are used the heuristic yields a cost metric that is no more than 5.7% greater than the optimal. This variation arises because whereas the optimization approach decided that it was best to deploy 12 sensors, the heuristic chose to deploy only 8 sensors. Hence, the heuristic yields a higher system cost metric due to the smaller number of sensors. For the nonlinear sensor cost model Fig. 6.2b shows reasonable agreement between the optimization and heuristic approaches. However, as the number of sensors available for use increases, the heuristic overestimates the system cost metric. As before, this is because the heuristic chooses to use fewer sensors than are necessary to achieve an optimal cost. This time the maximum variation between the heuristic and the optimization approaches is no more than 3.3%. This variation indicates that the heuristic performs better under the nonlinear sensor cost model.

For the train with 20 containers Fig. 6.3a shows that under the linear sensor cost model the optimization and heuristic approaches have reasonable agreement if 2 or 5 sensors. When more than 5 sensors are used the heuristic yields a cost metric that is no more than 5% greater than the optimal. As we saw for the 33 containers train, this arises because the heuristic chooses to deploy fewer sensors than are necessary to achieve the optimal system cost metric. For the nonlinear sensor cost model Fig. 6.3b shows close agreement between the optimization and heuristic approaches if 2 or 5 sensors are used. As was observed for the 33 containers train the heuristic overestimates the system cost metric as the number of sensors available for use increases. This time the maximum variation between the heuristic and optimization approaches is no more than 3.3%. This also shows indicates that the heuristic performs better under the nonlinear sensor cost model.

Finally, for the train with 14 containers Fig. 6.4a shows that under the linear sensor cost model the optimization and heuristic approaches show very good agreement if 2 sensors are assigned. In this case the heuristic decided that it was best to use 4 sensors while the optimal solution was 5 sensors. When more than 5 sensors are available for use the heuristic yields a cost metric that is no more than 7.2% greater than the optimal. For the nonlinear sensor cost model Fig. 6.4b shows good agreement between the heuristic and optimization approach if 2 sensors are assigned. As was the case for the linear sensor cost model the heuristic decided that it was best to deploy 4 sensors while the optimal solution was 5 sensors. As has been seen in every case so far the heuristic always overestimates the system cost metric. This time the maximum variation between the heuristic and optimization approaches is no more than 4%. From Fig. 6.2–6.4 it can be concluded that the heuristic produces results that are no more than 7.2% greater than the optimal solution. Therefore, the heuristic shows



(a) Train with 33 containers

(b) Train with 20 containers

Figure 6.5. Sensor Locations during Heuristic Validation

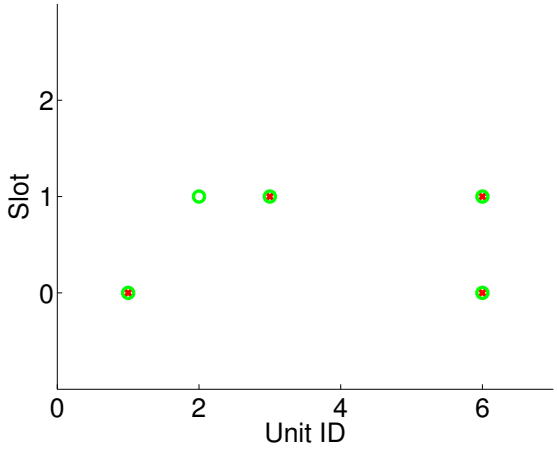


Figure 6.6. Sensor Locations for Train with 14 Containers

sufficient promise to be deployed to a real intermodal train.

Figs. 6.5 and 6.6 compare the sensor locations chosen by the heuristic and the optimization approaches. In Figs. 6.5 and 6.6 the red crosses show the best sensor locations identified by the heuristic during validation. The green circles on the other hand reflect the optimal sensor locations. Figs. 6.5 and 6.6 show that the heuristic does not identify as many sensor locations as the optimization approach.

6.3.2 Heuristic Application

In this section we present results showing how the heuristic was applied on a typical intermodal train. The following assumptions were made when applying the heuristic:

- The average train speed was 25 km/h, which is computed from [86, 87].
- The length of the rail trip was 1984 km, which is the distance from Laredo to Kansas City [87].
- There were 105 units and 225 containers on the train, with 30 20-feet, 186 40-feet, and 9 45-feet containers. Furthermore, 150 containers had a mean value of 20,000 units, 50 containers had a mean value of 100,000 units, and 25 containers had a mean value of 200,000 units. Recall, from Lai *et al.* [19] that an international intermodal stack train can have up to 104 units and 224 containers.
- Containers were placed in slots on the train using only the train company's loading rules [78]. The value of the containers was not used in container placement.
- The probability of a critical event, such as a container seal being opened, closed, or tampered with, occurring at each container was varied across the runs.
- The sensor manufacturer's lower and upper limits on probability of detection are 0.60 and 0.95, respectively. Similarly, the sensor manufacturer's lower and upper limits on probability of false alarm are 0.001 and 0.005, respectively.

The heuristic was run for the cases where there were between 3 and 225 sensors available to be placed on the containers. Fig. 6.7 is a partial copy of the heuristic

```
sensor 1 on railcar 4 and container 7
sensor 2 on railcar 25 and container 65
sensor 3 on railcar 92 and container 199
sensor 4 on railcar 86 and container 186
sensor 5 on railcar 21 and container 56
sensor 6 on railcar 78 and container 170
sensor 7 on railcar 46 and container 107
sensor 8 on railcar 50 and container 114
sensor 9 on railcar 2 and container 4
```

Figure 6.7. Partial Copy of Heuristic Output Showing Sensor Placement

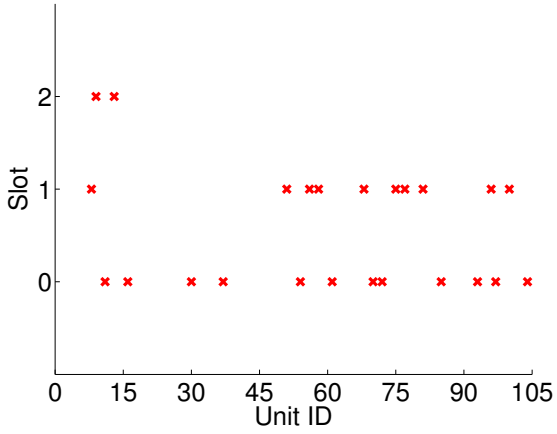
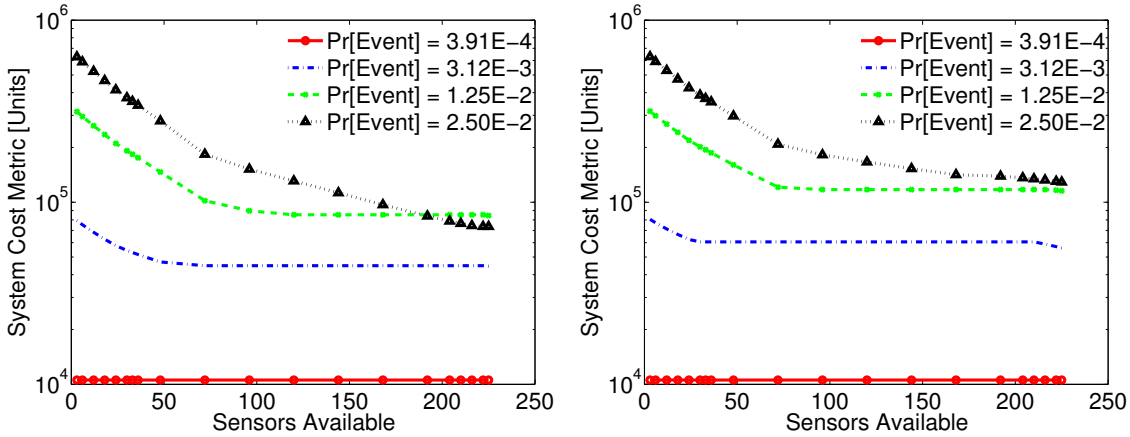


Figure 6.8. Sensor Locations for Train with 225 Containers



(a) Linear Sensor Cost (b) Nonlinear Sensor Cost

Figure 6.9. Application of the Sensor Assignment Heuristic

output showing sensor locations on railcars and containers. Fig. 6.8 shows the best sensor assignment pattern to the 225 container train when the probability of event occurrence is 3.125×10^{-3}. In Fig. 6.8 the locomotive is represented by unit 0, while slots 0 and 2 are in the bottom level of the railcar, and slot 1 is in the top level.

Fig. 6.9 shows the results of applying the heuristic to the 105 unit train for different probabilities of critical event occurrence. Fig. 6.9a shows that for the linear sensor cost model if the probability of critical event occurrence is greater than or equal to 0.0125 then the system cost metric decreases as more sensors are assigned. Recall from Fig. 6.1 that sensors are assigned to containers as long as the unit sensor cost is less than the savings that would be lost if an undetected event occurs, i.e.,

$$C_H < \sigma_j \zeta \tag{6.2}$$

The condition in equation (6.2) can result in a suboptimal stopping condition. When the probability of critical event occurrence is 0.0125, equation (6.2) allows sensors to be assigned to only 120 containers, while when the probability of critical event occurrence is 0.025 up to 225 sensors are assigned. When the probability of critical event occurrence is 3.125×10^{-3}, the best system cost metric, under the linear sensor cost model, is achieved if about 72 sensors are used. As more sensors are assigned they do not add any value to the system since the cost of the additional sensors is more than the savings that would be lost in undetected events. Finally, when the probability of critical event occurrence is 3.91×10^{-4} the unit sensor cost, which is at least 250 units, is always greater than the cost of undetected events. As a result, the heuristic does not deploy sensors in this case and the system cost metric reflects the cost of doing nothing.

Fig. 6.9b shows the results of applying the heuristic to a 225 container train with the nonlinear sensor cost model. When the probability of critical event occurrence is 3.125×10^{-3} the best system cost metric under the nonlinear sensor cost model is seen when about 30 sensors are deployed. The rest of the major trends from Fig. 6.9b match those seen in Fig. 6.9a, except that the system cost metric is higher under the nonlinear sensor cost model.

6.4 Conclusion

This chapter has presented a heuristic for assigning sensors to containers on typical intermodal trains. The results presented above show that on a sample train the heuristic's performance is comparable to the optimization-based approach for 14, 20, and 33 containers. Furthermore, the heuristic is able to determine appropriate sensor assignments and sensor characteristics for near to optimal system performance when applied to a typical international intermodal train. The heuristic has also shown that depending on the probability of critical event occurrence it may be necessary to use fewer sensors than the total number of containers, if the cost of an event exceeds the unit sensor cost.

The heuristic presented in this chapter is one step towards developing methods for placing sensors on containers on a train. The heuristic can also be applied to studying system trade-offs. As Fig. 6.9 showed it may sometimes be necessary to not cover every container with a sensor when monitoring cargo. As a result, a stopping condition has been included in the heuristic to stop assigning sensors if each additional sensor costs more than an undetected event. All other refinements to the heuristic are deferred to future work.

Chapter 7

Conclusions

The proposed approach for monitoring cargo in motion is the first work that the author is aware of that combines mathematical modeling and a testbed deployment of a sensor network for monitoring cargo. This chapter recounts the key findings of this dissertation and lists some opportunities for future work.

7.1 Lessons Learned

This section summarizes the key findings of this dissertation. The review of literature showed that optimization theory is being used to improve train operations. In addition, the solution of mixed-integer nonlinear programs, such as those in the proposed optimization models, was found to be $\mathcal{NP}$ (nondeterministic polynomial time) complete. The author found that wireless sensor networks are increasingly used for security monitoring. Furthermore, the literature review revealed that train equipment has been monitored using software built on a service-oriented architecture. In addition, security protocols have been developed for communicating with sensors inside shipping containers. However, the review of literature does not indicate that anyone has combined sensors and an open service-oriented architecture to monitor freight in motion. Finally, the literature review showed that no clear method for connecting trains to the global Internet has yet emerged.

Next, we studied two field trials of an open system transportation security sensor

network. Based on data collected from the field trials it was seen that decision makers could be notified of critical events, e.g., a seal being opened, closed, or tampered with, in about a minute. Further modeling work with the observed data showed that decision makers could be notified of critical events within 240 s with 99.9% probability. Thus, an open system transportation security sensor network could be used for monitoring cargo in motion.

Sensors can be used to provide visibility into cargo shipments. As a result, two models were developed—one for use when all network elements are on the train and the other for use when some are located trackside—to determine sensor placements and network design. The models show that, under reasonable assumptions, sensor deployment reduces the overall system cost; therefore, sensor networks make sense for monitoring cargo. The models also enabled the study of system trade-offs while achieving the desired level of visibility into cargo shipments.

The system trade-off studies showed that the system deployment cost is inversely related to the deadline for decision maker notification. The system deployment cost is also inversely related to the average train speed, with the trackside model being most greatly affected by the train speed. Thus, system designers deciding to implement either the train-mounted or trackside systems can trade-off train speed and event notification deadline with system cost, though in general the train companies would always go as fast as possible. Finally, the system trade-off studies showed that for the train-mounted system emphasis should be paid to the probability of false alarm before the probability of detection. For the trackside system more emphasis should be paid to the probability of false alarm, then the probability of successful communications, and then the probability of detection.

Since the optimization models result in mixed-integer nonlinear programs, which

are $\mathcal{NP}$-complete, the system trade-off studies were based on a relatively small train with 15 units and 33 containers. However, Lai *et al.* [19] state that an international intermodal stack train can have up to 104 units and 224 containers. Thus, a heuristic was developed to deploy sensors on typical intermodal trains. The results presented in Chapter 6 show that on a sample train of 105 units and 225 containers, the heuristic's performance is comparable to the optimization-based approach.

7.2 Future Work

Section 7.1 summarized the key findings of this dissertation. This section discusses some opportunities for future work based on this dissertation.

- In Chapter 4 the sensor assignment models were "validated" by studying trends in their behavior especially at the boundaries of the visibility space. In the future one needs to investigate more rigorous ways of validating the models.
- Model enhancements:
  - In Sections 5.3.6 and 5.4.7 the effects of variation the probability of detection were studied. It is reasonable to assume that sensors with smaller variation in the probability of detection are better than those with larger variation. Thus, can the sensor cost models be improved such that they incorporate the standard deviation in sensor characteristics?
  - The models proposed in this dissertation assume that the train operator sees no loss if a critical event is detected and reported in a timely manner. However, it is possible that the train operator may see a small loss even if the event is detected and reported in a timely manner. Thus, the

calculation for the system cost metric should be enhanced such that it incorporates a small loss to the system operator if there is an event.

- The literature review indicates that cargo is most at risk when it is stationary. Thus, the probability of a critical event needs to be correlated with train speed. It is also reasonable to assume that there is positive correlation between train speed and the probability of false alarm. Thus, the model should be improved to handle this case.
- In Section 5.6 we observed that the models cannot currently handle the case where the backhaul communications device contacts the operations center periodically. The models need to be extended to handle this case.
- Section 4.6.1 showed that there was a very rapid growth in the number of variables in the models. Part of this growth is an artifact of the binary variable used to indicate sensor placement. Thus, we need to determine if a more concise model will allow for the solving of larger trains. In addition, we will investigate if the sensor placement problem on larger trains can be solved by using parallel processing.
- Other methods of visualizing the results of this study need to be investigated. The new results visualization approach should clearly show the benefits of deploying a cargo monitoring system.
- Finally, the part of the model that incorporates timely notification of events needs to be improved. In the short-haul trial, the HSDPA link from the train to the operations center was always up. However, in the long-haul trial the Iridium link to the virtual network operations center was established periodically. Thus, the model needs to factor into the notification time the period when there is no stable connection to report events.

- Other system enhancements:
  - The hardware used in the TSSN field trials did not support ad hoc networking of the sensor nodes. It would be possible to monitor longer trains if the sensor nodes could form an ad hoc network. Two tasks need to be accomplished to do this: 1) redesigning the hardware for the sensors to accomplish this task; and 2) designing software on the sensors to allow for forwarding of packets to the appropriate sensor.

References

[1] Federal Bureau of Investigation. (2006, Jul. 21) Cargo Theft's High Cost. Headline. Federal Bureau of Investigation. [Online]. Available: http://www.fbi.gov/page2/july06/cargo_theft072106.htm

[2] C. Mayhew, "The Detection and Prevention of Cargo Theft," *Trends & Issues in Crime and Criminal Justice*, no. 214, pp. 1–6, Sep. 2001. [Online]. Available: http://www.aic.gov.au/documents/B/B/0/%7BBB0D4DB9-5290-46E5-8438-486632808090%7Dti214.pdf

[3] European Conference of Ministers of Transport, *Container Transport Security Across Modes*. Paris, France: Organisation for Economic Co-operation and Development, 2005.

[4] KC SmartPort. (2008, Nov. 10) Trade Data Exchange—Nothing short of a logistics revolution. Digital magazine. [Online]. Available: http://www.joc-digital.com/joc/20081110/?pg=29

[5] R. Macmanus. (2009, Dec. 4) FedEx Joins the Internet of Things With SenseAware. News story. ReadWriteWeb. [Online]. Available: http://www.nytimes.com/external/readwriteweb/2009/12/04/04readwriteweb-fedex-joins-the-internet-of-things-with-sen-28351.html?emc=eta1

[6] Science Applications Int'l Corp., "KC SmartPort ITS Integration Project - Final Report," Draft, Apr. 2006.

[7] ——, "KC SmartPort ITS Economic Assessment," Final, Apr. 2006.

[8] J. H. Armstrong, *The Railroad, What It Is, What It Does: The Introduction to Railroading*, 4th ed. Omaha, NE, USA: Simmons-Boardman Books, 1998.

[9] P. Fuhr and R. Lau. (2005, Mar. 1) Mesh Radio Network Performance in Cargo Containers. Article. [Online]. Available: http://mil.sensorsmag.com/sensorsmil/Emerging+Technologies/Mesh-Radio-Network-Performance-in-Cargo-Containers/ArticleStandard/Article/detail/270167

[10] J. L. Schoeneman *et al.*, "WIPP Transparency Project-Container Tracking and Monitoring Demonstration using the Authenticated Tracking and Monitoring System (ATMS)," presented at the Waste Management 2000 (WM2K) Conf., Tucson, AZ, USA, Feb. 2000. [Online]. Available: http://www.osti.gov/bridge/product.biblio.jsp?osti_id=751077

[11] J. Ove Lauf and H. Sauff, "Secure Lightweight Tunnel for Monitoring Transport Containers," in *Proc. 3rd Int'l Conf. Security and Privacy Communications Networks (SecureComm 2007)*, Nice, France, Sep. 2007, pp. 484–493.

[12] C. Decker *et al.*, "Cost-Benefit Model for Smart Items in the Supply Chain," in *Proc. 1st Int'l Conf. The Internet of Things (IOT 2008)*, ser. Lecture Notes in Computer Science, vol. 4952/2008. Springer Berlin / Heidelberg, Mar. 2008, pp. 155–172.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering, T. V. Mikosch *et al.*, Eds. Springer, 2006.

[14] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY USA: Dover Publications, 1998.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[16] A. N. Letchford, "Mixed-Integer Non-Linear Programming: A Survey," presented at the 1st LANCS Workshop Discrete and Non-Linear Optimisation, Southampton, UK, Feb. 2009.

[17] K. Darby-Dowman and J. M. Wilson, "Developments in Linear and Integer Programming," *The Journal of the Operational Research Society*, vol. 53, no. 10, pp. 1065–1071, Oct. 2002. [Online]. Available: http://www.jstor.org/stable/822966

[18] P. Bonami *et al.*, "Algorithms and Software for Convex Mixed Integer Nonlinear Programs," Computer Sciences Department, University of Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1664, Oct. 2009.

[19] Y.-C. Lai *et al.*, "Optimizing the Aerodynamic Efficiency of Intermodal Freight Trains," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 5, pp. 820–834, Sep. 2008.

[20] Y. Ouyang *et al.*, "Optimal Locations of Railroad Wayside Defect Detection Installations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 5, pp. 309–319, 2009.

[21] A. Billionnet, "Using Integer Programming to Solve the Train-Platforming Problem," *Transportation Science*, vol. 37, no. 2, pp. 213–222, 2003.

[22] J.-F. Cordeau *et al.*, "A Survey of Optimization Models for Train Routing and Scheduling," *Transportation Science*, vol. 32, no. 4, pp. 380–404, Apr. 1998.

[23] T. A. Feo and J. L. González-Velarde, "The Intermodal Trailer Assignment Problem," *Transportation Science*, vol. 29, no. 4, pp. 330–341, 1995.

[24] W. B. Powell and T. A. Carvalho, "Real-Time Optimization of Containers and Flatcars for Intermodal Operations," *Transportation Science*, vol. 32, no. 2, pp. 110–126, 1998.

[25] N. Bostel and P. Dejax, "Models and Algorithms for Container Allocation Problems on Trains in a Rapid Transshipment Shunting Yard," *Transportation Science*, vol. 32, no. 4, pp. 370–379, Nov. 1998.

[26] P. Corry and E. Kozan, "An Assignment Model for Dynamic Load Planning of Intermodal Trains," *Comput. Oper. Res.*, vol. 33, no. 1, pp. 1–17, 2006.

[27] F. Van Quickenborne *et al.*, "Optimization Models for Designing Aggregation Networks to Support Fast Moving Users," in *Proc. 1st Int'l Workshop EURO-NGI Network of Excellence on Wireless Systems and Mobility in Next Generation Internet*, G. Kotsis and O. Spaniol, Eds., vol. LNCS 3427. Dagstuhl, Germany: Springer, Jun. 2005, pp. 66–81.

[28] G. M. Shafiullah *et al.*, "Survey of Wireless Communications Applications in the Railway Industry," in *Proc. 2nd Int'l Conf. Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, Sydney, NSW, Australia, Aug. 2007, pp. 65–70.

[29] M. C. Edwards *et al.*, "Improving Freight Rail Safety with on-board Monitoring and Control Systems," in *Proc. ASME/IEEE Joint Rail Conf.*, Pueblo, CO, USA, Mar. 2005, pp. 117–122.

[30] Q. Shan *et al.*, "Wireless Intelligent Sensor Networks for Refrigerated Vehicle," in *Proc. IEEE 6th Circuits and Systems Symp. Emerging Technologies: Frontiers of Mobile and Wireless Communication*, vol. 2, Shanghai, China, May 2004, pp. 525–528.

[31] L. Ruiz-Garcia *et al.*, "Review. Monitoring the intermodal, refrigerated transport of fruit using sensor networks," *Spanish Journal of Agricultural Research*, vol. 5, no. 2, pp. 142–156, 2007.

[32] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[33] D. T. Fokum *et al.*, "A Taxonomy of Sensor Network Architectures," University of Kansas, Lawrence, KS, Tech. Rep. ITTC-FY2008-TR-41420-07, Jan. 2008.

[34] S. Avancha *et al.*, "Secure Sensor Networks for Perimeter Protection," *Computer Networks*, vol. 43, no. 4, pp. 421–435, Nov. 2003.

[35] E. Onur *et al.*, "Finding Sensing Coverage and Breach Paths in Surveillance Wireless Sensor Networks," in *Proc. 15th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, vol. 2, Barcelona, Spain, Sep. 2004, pp. 984–988.

[36] ——, "How Many Sensors for an Acceptable Breach Detection Probability?" *Computer Communications*, vol. 29, no. 2, pp. 173–182, Jan. 2006.

[37] L. W. Krakow *et al.*, "Control of Perimeter Surveillance Wireless Sensor Networks via Partially Observable Marcov Decision Process," in *Proc. 40th Annual IEEE Int'l Carnahan Conf. Security Technology.* Lexington, KY, USA: IEEE, Oct. 2006, pp. 261–268.

[38] A. Efrat *et al.*, "Approximation Algorithms for Two Optimal Location Problems in Sensor Networks," in *Proc. 2nd Int'l Conf. Broadband Networks (BROADNETS 2005)*, Boston, MA, USA, Oct. 2005, pp. 714–723.

[39] K. Chakrabarty *et al.*, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[40] K. Harman and A. Gagnon, "Synergistic Radar: a new Approach to Intrusion Detection," in *Proc. IEEE Int'l Carnahan Conf. Security Technology, Crime Countermeasures*, Atlanta, GA, USA, Oct. 1992, pp. 8–13.

[41] T. L. Bisbee and D. A. Pritchard, "Today's Thermal Imaging Systems: Background and Applications for Civilian Law Enforcement and Military Force Protection," in *Proc. IEEE 31st Annual Int'l Carnahan Conf. Security Technology*, Canberra, Australia, Oct. 1997, pp. 202–208.

[42] M. Horner *et al.*, "AMETHYST: Automatic Alarm Assessment Becoming a Reality," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 13, no. 7, pp. 31–36, Jul. 1998.

[43] T. Bass, "Intrusion Detection Systems and Multisensor Data Fusion," *Commun. ACM*, vol. 43, no. 4, pp. 99–105, Apr. 2000.

[44] C. J. Tarr, "Cost Effective Perimeter Security," in *Proc. European Convention Security and Detection*, Brighton, UK, May 1995, pp. 183–187.

[45] S. H. Jacobson *et al.*, "A Detection Theoretic approach to Modeling Aviation Security Problems using the Knapsack Problem," *IIE Transactions*, vol. 33, no. 9, pp. 747–759, Sep. 2001.

[46] A. Khalafallah and K. El-Rayes, "Minimizing Construction-Related Security Risks during Airport Expansion Projects," *Journal of Construction Engineering and Management*, vol. 134, no. 1, pp. 40–48, Jan. 2008. [Online]. Available: http://link.aip.org/link/?QCO/134/40/1

[47] D. T. Fokum and V. S. Frost, "A Survey on Methods for Broadband Internet Access on Trains," University of Kansas, Lawrence, KS, Tech. Rep. ITTC-FY2008-TR-41420-09, Aug. 2008.

[48] ——, "A Survey on Methods for Broadband Internet Access on Trains," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 171–185, Quarter 2 2010.

[49] D. T. Fokum *et al.*, "Experiences from a Transportation Security Sensor Network Field Trial," University of Kansas, Lawrence, KS, Tech. Rep. ITTC-FY2009-TR-41420-11, Jun. 2009.

[50] ——, "An Open System Transportation Security Sensor Network: Field Trial Experiences," *Trans. Veh. Technol., IEEE*, vol. 59, no. 8, pp. 3942–3955, Oct. 2010.

[51] J. Fernandez *et al.*, "Transf-ID: Automatic ID and Data Capture for Rail Freight Asset Management," *IEEE Internet Comput.*, vol. 13, no. 1, pp. 22–30, Jan. 2009.

[52] M. Kuehnhausen, "Service Oriented Architecture for Monitoring Cargo in Motion Along Trusted Corridors," Master's thesis, University of Kansas, Jul. 2009.

[53] D. T. Fokum *et al.*, "Experiences from a Transportation Security Sensor Network Field Trial," in *Proc. 3rd IEEE Workshop Enabling the Future Service-Oriented Internet: Towards Socially-Aware Networks (EFSOI 2009).* Honolulu, HI: IEEE, Dec. 2009, pp. 1–6.

[54] ——, "An Open System Transportation Security Sensor Network: Field Trial Experiences," University of Kansas, Lawrence, KS, USA, Tech. Rep. ITTC-FY2010-TR-41420-21, Mar. 2010.

[55] OASIS. (2006, Oct. 12) Reference Model for Service Oriented Architecture 1.0. OASIS Standard. [Online]. Available: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[56] Kansas City Southern Railroad, Private communication, 2007.

[57] A. Arsanjani *et al.*, "Web Services: Promises and Compromises," *Queue*, vol. 1, no. 1, pp. 48–58, Mar. 2003.

[58] H. Saiedian and S. Mulkey, "Performance Evaluation of Eventing Web Services in Real-time Applications," *IEEE Commun. Mag.*, vol. 46, no. 3, pp. 106–111, Mar. 2008.

[59] J. Brown *et al.*, "SMS: The Short Message Service," *Computer*, vol. 40, no. 12, pp. 106–110, Dec. 2007.

[60] R. Chinnici *et al.* (2007, Jun.) Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation. W3C. [Online]. Available: http://www.w3.org/TR/wsdl20/

[61] M. Gudgin *et al.* (2007, Apr.) SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Member submission. W3C. [Online]. Available: http://www.w3.org/TR/soap12-part1/

[62] D. Box *et al.* (2004, Aug. 10) Web Services Addressing (WS-Addressing). Member submission. W3C. [Online]. Available: http://www.w3.org/Submission/ws-addressing/

[63] OASIS. (2004, Mar.) Web Services Security: SOAP Message Security 1.0. OASIS Standard. OASIS. [Online]. Available: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[64] D. Box *et al.* (2006, Mar.) Web Services Eventing (WS-Eventing). Member Submission. W3C. [Online]. Available: http://www.w3.org/Submission/WS-Eventing/

[65] The Apache Software Foundation. (2008, Aug. 24) Apache Axis2. Project documentation. The Apache Software Foundation. [Online]. Available: http://ws.apache.org/axis2/

[66] D. Mulvey, "HSPA," *Communications Engineer*, vol. 5, no. 1, pp. 38–41, Feb.-Mar. 2007.

[67] C. E. Fossa *et al.*, "An overview of the IRIDIUM (R) low Earth orbit (LEO) satellite system," in *Proc. IEEE Nat'l Aerospace and Electronics Conf. (NAECON 1998)*, Dayton, OH, USA, Jul. 1998, pp. 152–159.

[68] Hi-G-Tek. (2009, Mar. 17) Hi-G-Tek—Company. Corporate website. Hi-G-Tek. [Online]. Available: http://www.higtek.com/

[69] The Apache Software Foundation. (2007, Sep. 1) Apache log4j. Project documentation. The Apache Software Foundation. [Online]. Available: http://logging.apache.org/log4j/

[70] EsperTech. (2009, Feb. 11) Esper – Complex Event Processing. Project documentation. EsperTech. [Online]. Available: http://esper.codehaus.org/

[71] *DataReader and DataSeal : User's Manual*, UM4710, Hi-G-Tek Ltd., 2001, ver. A5.

[72] Google. (2009, May 6) Google Maps. Web Mapping Service. [Online]. Available: http://maps.google.com

[73] D. L. Mills, "Internet Time Synchronization: the Network Time Protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[74] J. Clark and S. DeRose. (1999, Nov. 16) XML Path Language (XPath). W3C Recommendation. W3C. [Online]. Available: http://www.w3.org/TR/xpath

[75] D. T. Fokum, "Optimal Communications Systems and Network Design for Cargo Monitoring," Presented at the 10th Workshop Mobile Computing Systems and Applications (HOTMOBILE 2009). Santa Cruz, CA, USA: ACM, Feb. 2009.

[76] S. Nadarajah, "A Review of Results on Sums of Random Variables," *Acta Applicandae Mathematicae*, vol. 103, no. 2, pp. 131–140, Sep. 2008.

[77] E. Komp *et al.*, "Implementing Web Services: Conflicts Between Security Features and Publish/Subscribe Communication Protocols," University of Kansas, Lawrence, KS, Tech. Rep. ITTC-FY2010-TR-41420-19, Feb. 2010.

[78] Intermodal Committee, *Loading Capabilities Guide*, Association of American Railroads Std., Jun. 26 2003. [Online]. Available: http://www.aar.org/AARPublications/~/media/AARPublications/FreePubs/AAR%20Loading%20Capabilities%20Guide.ashx

[79] G. B. Kleindorfer *et al.*, "Validation in Simulation: Various Positions in the Philosophy of Science," *Manage. Sci.*, vol. 44, no. 8, pp. 1087–1099, Aug. 1998.

[80] P. Bonami. (2010, May) Bonmin. Project wiki. [Online]. Available: https://projects.coin-or.org/Bonmin

[81] (2010, Feb.) NEOS Solvers. Solver listing. Argonne National Labs. Argonne, IL, USA. [Online]. Available: http://neos.mcs.anl.gov/neos/solvers/index.html

[82] J. Czyzyk *et al.*, "The NEOS Server," *IEEE Comput. Sci. Eng.*, vol. 5, no. 3, pp. 68–75, Jul.-Sep. 1998.

[83] S. E. Flynn. (2006, Jan.–Feb.) Port Security Is Still a House of Cards. Article. Council on Foreign Relations. [Online]. Available: http://www.cfr.org/publication/9629/

[84] M. Wolfe. (2002, Apr.) Technology to Enhance Freight Transportation Security and Productivity. Appendix to: "Freight Transportation Security and Productivity". [Online]. Available: http://ops.fhwa.dot.gov/freight/publications/sec_tech_appx/security_tech_appx.htm

[85] M. Kuehnhausen and V. S. Frost, "Application of the Java Message Service in Mobile Monitoring Environments," University of Kansas, Lawrence, KS, USA, Tech. Rep. ITTC-FY2010-TR-41420-18, Dec. 2009.

[86] Kansas City Southern Railroad, Intermodal shipping schedule, Jun. 2007. [Online]. Available: http://www.kcsouthern.com/en-us/Customers/Documents/Kansas%20City%20Southern%20Intermodal%20Schedules.xls

[87] ——, System map, Jun. 2010. [Online]. Available: http://www.kcsi.com/SiteCollectionDocuments/system_map.pdf

[88] (2010, Jun. 11) Available Data Rate Plans. Price schedule. AT & T. [Online]. Available: http://www.wireless.att.com/businesscenter/popup/dataconnect-comp-table.jsp

[89] (2010, Jun.) NAL - Airtime. Price schedule. NAL Research Corporation. [Online]. Available: http://www.nalresearch.com/Airtime.html#Dial-Up%20Data

[90] P. M. Frank, *Introduction to System Sensitivity Theory.* New York, NY, USA: Academic Press, 1978.

[91] A. M. Breipohl, *Probabilistic Systems Analysis: An Introduction to Probabilistic Models, Decisions, and Applications of Random Processes.* John Wiley & Sons, Inc., 1970.