Experiences from a Transportation Security Sensor Network Field Trial

Daniel T. Fokum*†, Victor S. Frost*, Daniel DePardo*, Martin Kuehnhausen*, Angela N. Oguna*,
Leon S. Searl*, Edward Komp*, Matthew Zeets*, Daniel Deavours*, Joseph B. Evans*, and Gary J. Minden*
*Information and Telecommunication Technology Center,
The University of Kansas,
Lawrence, KS, 66045, USA
†Corresponding author: fokumdt@ittc.ku.edu

***Abstract*—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. Prior to deploying the TSSN it should be determined if the system can provide timely event notification. A field experiment was conducted to assess the TSSN's suitability for monitoring rail-borne cargo. Log files were collected from this experiment and postprocessed. In this paper we present empirical results on the time taken to report events using the TSSN. These results show that the TSSN can be used to monitor rail-borne cargo.**

***Index Terms*—Service oriented architecture, Mobile Rail Network, Trade Data Exchange, Virtual Network Operations Center**

I. Introduction

In 2006 the FBI estimated that cargo theft cost the US economy between 15 and 30 billion dollars per year [1]. Cargo theft affects originators, shippers, and receivers as follows: originators need a reliable supply chain to deliver goods in a timely and cost-effective manner (A receiver's ability to receive goods in a timely manner affects the originator.). Shippers, on the other hand, hold liability and insurance costs for shipments and these costs are proportional to the rate of theft. Finally, receivers are impacted by out-of-stock and scheduling issues due to cargo theft. Most non-bulk cargo travels in shipping containers. Container transport is characterized by complex interactions between shipping companies, industries, and liability regimes [2]. Deficiencies in the container transport chain expose the system to attacks such as the Trojan horse (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 shipment information. One crucial research question that must be answered in order to attain this objective is how to create technologies that will allow continuous monitoring of containers by integrating commodity 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.

To achieve improved efficiency and security of trade lanes, we have developed a Transportation Security Sensor Network (TSSN), based on Service Oriented Architecture (SOA) [3] principles, for monitoring the integrity of rail-borne cargo shipments. 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 II. The TSSN detects events and reports those important to decision makers using commodity networks. Ideally, decision makers would be notified of events within 15 minutes so that they can take effective action. For the TSSN to be deployed we need to understand the timeliness of the system response; however, we do not know *a priori* how the TSSN would perform due to the unknown execution time of SOA-based programs ([5] and [6]), unpredictable packet latency on commodity networks, and the use of email and/or SMS (Short Message Service) [7] for alarm notification. Thus, we have carried out an experiment to characterize the TSSN system, particularly the end-to-end time between event occurrence and decision maker notification using SMS.

This paper presents a description of our cargo monitoring system and experimental results showing the time taken to notify shippers of events on a train. These results indicate that decision makers can be notified of events on the train in a timely manner using the TSSN. The rest of this paper

This work was supported in part by Oak Ridge National Laboratory (ORNL)—Award Number 4000043403. This material is also partially based upon work supported while V.S. Frost was serving at the National Science Foundation.

is laid out as follows: In Section II we present a description of the TSSN system architecture including the components. Section II also discusses the hardware configuration used in the MRN. In Section III we discuss an experiment conducted to assess the suitability of the TSSN system for cargo monitoring. Section IV presents empirical results showing TSSN performance. In Section V we discuss refinements to the TSSN architecture based on preliminary results. Section VI provides concluding remarks.

II. System Architecture

To achieve the objectives presented in Section I we have designed and implemented a system called the Transportation Security Sensor Network (TSSN). The detailed architecture of the TSSN is found in [8], whereas this section gives an overview of the TSSN. The architectural details presented here are important in understanding the experiment and results presented in Sections III and IV, 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, 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), Simple Object Access Protocol (SOAP 1.2), WS-Addressing, WS-Security, and WS-Eventing—which are implemented through Apache Axis2 [9] and associated modules. These standards are used to exchange structured information between a web service and client. The use of SOAP allows the use 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 processing overhead related to SOAP messages.

The TSSN supports wireless and satellite communication technologies such as HSDPA (High-Speed Downlink Packet Access) [10] and Iridium [11]. The TSSN uses the Hypertext Transfer Protocol (HTTP) for message transport over wired and wireless links. Finally, the current TSSN prototype uses sensors and readers from Hi-G-Tek [12]. There is also a need to gather log files to enable system debugging as well 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 [13].

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. 1. Wired links are used between the TDE and the VNOC, while MRN to VNOC communications are done using wireless links. The TDE, VNOC, and MRN are examined in greater detail in the following subsections.

A. 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 it responds to queries from the VNOC. The TDE also stores alarm 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.

B. Virtual Network Operations Center

The Virtual Network Operations Center (VNOC) is the shipper's interface to the TDE. The VNOC is also the central decision and connection point for all of a shipper's MRNs. The VNOC runs on a server with a wired connection to the Internet and it performs the following functions:

- Receives messages from the MRN.
- Obtains event-associated cargo information from the Trade Data Exchange (TDE).
- Makes decisions (using rules) on which MRN alarms are ignored or forwarded to decision makers, for example, a low battery alarm is sent to technical staff while an open/close event is sent to decision makers.
- Combines cargo information obtained in real time from the TDE with an MRN alarm to form a VNOC alarm message that is sent (by SMS and/or email) to decision makers.
- Forwards commands from a TDE client to the TSSN collector node to start and stop monitoring at the MRN, as well as to get the MRN's current location.

C. Mobile Rail Network

The MRN subsystem consists of hardware and software. We discuss the hardware and software architecture below.

1) Mobile Rail Network Hardware: The MRN subsystem hardware consists of a set of wireless shipping container security sensors and a TSSN collector node. Container physical security is monitored using a system that was originally designed for tanker truck security [12]. An interrogation transceiver communicates with active and battery-powered wireless data seals (sensors) over a wireless network. The interrogation transceiver communicates with a notebook computer via a serial data connection. The container seals are equipped with flexible wire lanyards that are threaded through container keeper bar lock hasps.

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. System communications using the Iridium modem are in the process of being implemented. The Iridium modem is a combination unit that includes a GPS receiver, which is used to provide the MRN position information.

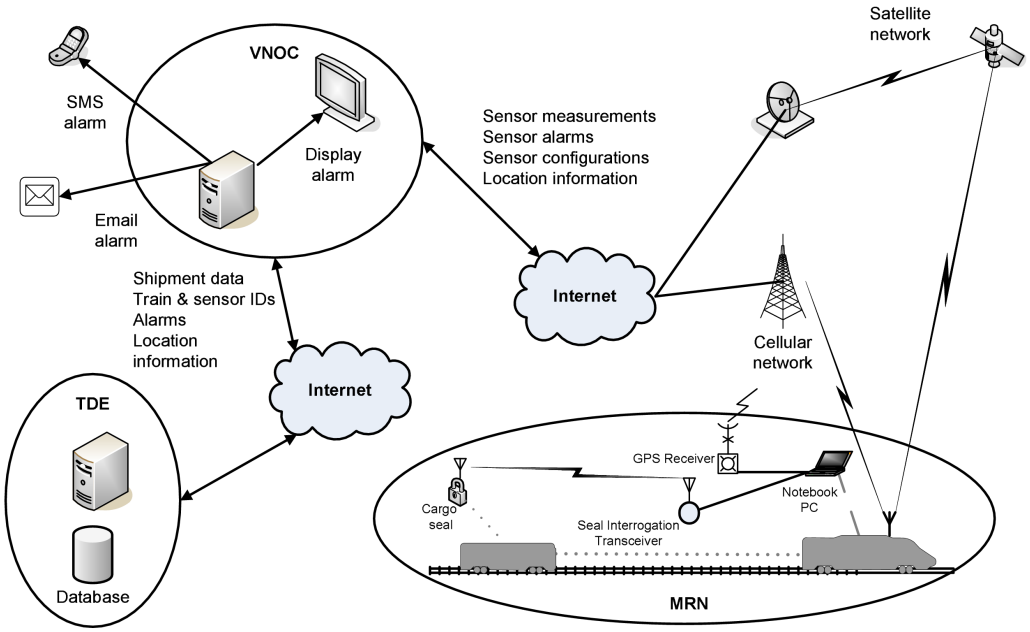

Fig. 1. Transportation Security Sensor Network (TSSN) Architecture

2) Mobile Rail Network Software: The MRN software consists of a SensorNode service, an AlarmProcessor service, and a Communications service. The SensorNode service finds and monitors sensors which 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. The interrogation transceiver also queries the seals at regular intervals. 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 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 currently logs the HSDPA signal strength. In the future we plan to build some intelligence into the Communication service so that it can switch between an Iridium and an HSDPA signal.

III. Experiment

We have conducted an experiment to assess the suitability of the TSSN system for cargo monitoring as well as to collect data that would be used to guide the design of future cargo monitoring systems. In this section we present the experimental objectives and set-up, data collected during the test, and issues that were encountered during the test.

A. Short-haul Rail Trial

This experiment was carried on a train making an approximately 35 km (22 miles) trip from an intermodal facility to a rail yard. Our objectives in this experiment were the following:

- To determine the performance of the TSSN system when detecting events on intermodal containers in a rail environment.
- To investigate if decision makers could be informed of events in a timely manner 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.

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 placed on intermodal shipping containers and in the locomotive.

During the experiment, events were created by breaking and closing a seal (sensor) that was kept in the locomotive. The VNOC reported these events to decision makers using e-mail and SMS messages. The e-mail messages also include a link to Google Maps, so that the exact location of the incident can be visualized. Fig. 2 shows the content of one of the e-mail messages that was sent to the decision makers and Fig. 3 presents an example of an SMS message.

Following this experiment, analysis of event logs generated on the MRN and VNOC revealed that there was a significant amount of clock drift on the TSSN Collector Node during this relatively short (about 5 hours) trial. The time recorded at the VNOC for receipt of a message, in some cases, was

```
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
```

Fig. 2. Email Message Sent During Short-haul Trial

```
NOC_Alarm:
Time:2009.07.04 14:16:36 CDT
GPS:38.95205/-95.26383
Trn:FS9999
Sev:Information
Type:SensorLimitReached
Msg:Sensortype=AVL SensorID=HGT18800977279
    Event=AllMissing
```

Fig. 3. SMS Message Generated by TSSN

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) [14] server, we conclude that the clock drift is occurring on the TSSN Collector Node. In the next version of the TSSN we have resolved the clock drift problem through a combination of software and hardware. 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 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 TSSN performance.

During the test the reader lost communication with the seals for a brief period along the route. Future experiments will determine whether or not this loss of connectivity was due to RF interference. In spite of this, the experiment 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 are being postprocessed to obtain data on TSSN system performance.

IV. Results

In this section we discuss the results of the TSSN system evaluation based on the short-haul rail trial. One objective of our experiments was to determine whether decision makers could be notified of events in a timely manner. Thus, we provide statistics on the end-to-end time between event occurrence and decision maker notification. More detailed statistics, including histograms, are found in [15].

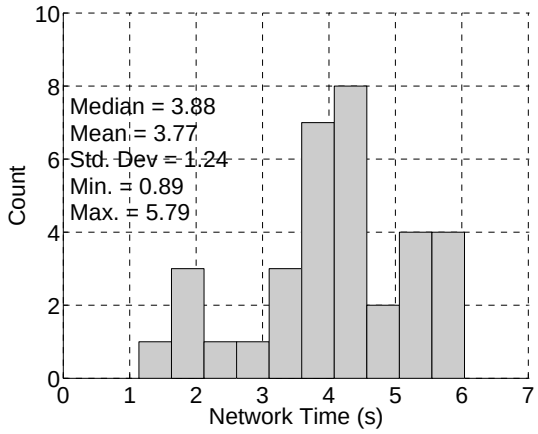

Fig. 4. Network Times from VNOC → MRN → VNOC

A. 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 requests include instructing the MRN to start or stop monitoring and getting the MRN's current location. These statistics allow us to gain insight into the one-way network delay from the TSSN collector node to the VNOC—a delay that is one component of sending an event report 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 for sending an MRN_Alarm message—which indicates an event at a sensor—to the VNOC. However, it is reasonable to assume that the MRN ↔ VNOC links are symmetric thus, the average one-way delay from the MRN to the VNOC is approximately 1.89 s.

B. Elapsed Time from Alert Generation to AlarmReporting Service

The time taken for the TSSN to process an event report is an important metric in evaluating this system. Furthermore, demonstrating that this metric is of the order of several seconds can help convince decision makers of the TSSN's utility. Fig. 5 shows the messages involved in notifying a decision maker of an event at a seal. 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 examining the log files from the experiment 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, we conclude that the time taken to process events in the TSSN is not an impediment to timely notification of decision makers.

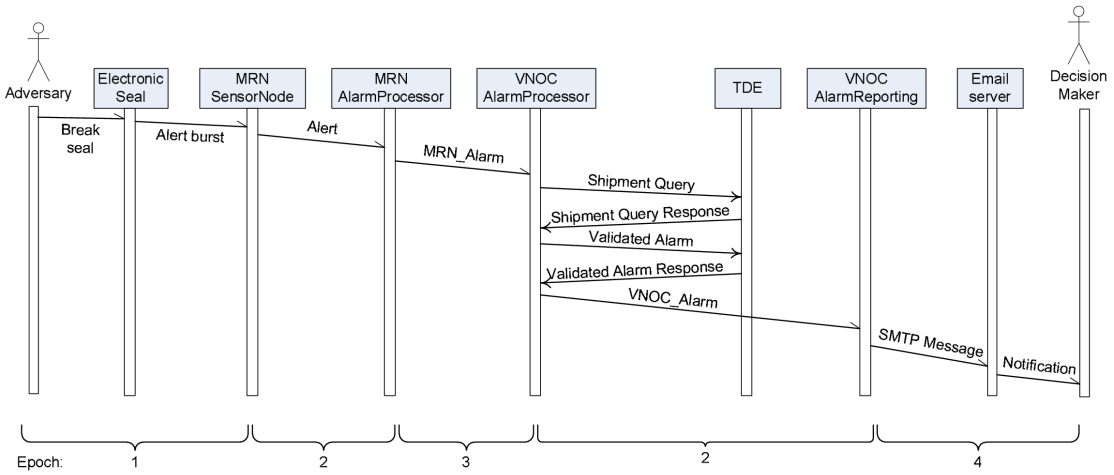

Fig. 5. Sequence Diagram with Messages Involved in Decision Maker Notification

TABLE I
SUMMARY OF TIME STATISTICS

Epoch	Description	Min./s	Max./s	Mean/s	Median/s	Std. Dev./s
1	Event occurrence to alert generation	0.81	8.75	2.70	2.13	1.86
2	Alert generation to VNOC AlarmReporting Service	1.92	4.91	2.08	1.97	0.32
3	One-way delay from MRN to VNOC	0.45	2.90	1.89	1.94	0.62
4	Delivery time from VNOC AlarmReporting Service to mobile phone	5.2	58.7	11.9	9.8	7.4

C. End-to-end Time from Event Occurrence to Decision Maker Notification

An important metric for TSSN performance is the time between event occurrence until a decision maker is notified using an SMS message. The components of the end-to-end time include:

- Time between between event occurrence and when the MRN SensorNode service generates the related event alert.
- Time from alert generation to the VNOC AlarmReporting service.
- Time taken for the VNOC AlarmReporting service to process and send an e-mail message to an e-mail server.
- Time taken by the SMS vendor to get the message to a decision maker's phone.

To 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 event alert the stopwatch was stopped. From Table I we see that the longest observed time between event occurrence and the MRN generating an Alert is about 8.8 s. Furthermore, it takes about 2.7 s on average.

A second experiment was carried out to determine the elapsed time between the VNOC AlarmReporting service's transmission of a VNOC alarm message and the decision maker receiving event notification. In this experiment a client program was written to send messages to the VNOC alarm reporting 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. Table I summarizes the statistics for delivery of alarm messages for four different carriers.

From Table I we see that even though SMS was not designed as a real-time system, it provides excellent notification for our purposes; since most of our messages were delivered within one minute. Combining all of these experimental results, we see that in the longest observed case it can take just over one minute[1] 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.

V. REFINEMENTS BASED ON EXPERIMENTAL RESULTS

Postprocessing 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.

[1]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.

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.
- Adding intelligence to the MRN software subsystem so that it can switch between satellite and cellular connections automatically based on signal availability.
- Enhancing sensor capabilities so that senors can communicate with each other 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.

VI. Conclusion

In this paper we have presented results from a field trial of the TSSN (Transportation Security Sensor Network). Within the TSSN framework we have successfully combined sensor and shipment information to provide event notification to distributed decision makers. This paper has shown results documenting the interactions between the different components of the TSSN. Based on our experiments and evaluations the TSSN is viable for monitoring rail-borne cargo. We have successfully demonstrated that alert messages can be sent from a moving train to geographically distributed decision makers using either SMS or e-mail. Furthermore, decision makers would like to get event notification within 15 minutes and our experimental results show that we are able to detect events and notify decision makers in just over one minute. Thus, we conclude that the TSSN provides a mechanism for timely notification of decision makers.

Acknowledgments

The authors would like to thank A. Francis reading and commenting on previous versions of this paper. The authors wish to acknowledge Kansas City Southern Railway for their participation in the short-haul rail trial. We would also like to acknowledge the support of EDS, an HP company, and Kansas City SmartPort, Inc. our partners on this project. Finally, L. Sackman of EDS, an HP company, assisted with the short-haul rail trial.

References

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

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

[3] 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

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

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

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

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

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

[10] D. Mulvey, "HSPA," *Communications Engineer*, vol. 5, no. 1, pp. 38–41, February-March 2007.

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

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

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

[14] D. L. Mills, "Internet Time Synchronization: the Network Time Protocol," *Communications, IEEE Transactions on*, vol. 39, no. 10, pp. 1482–1493, Oct 1991.

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