A Functionality Based Hierarchical Model for Survivable Intelligent Transportation Systems*

Axel W. Krings† and Patrick R. Merry
Computer Science Department
University of Idaho
Moscow, ID 83844–1010
{krings,patrickm}@cs.uidaho.edu

Abstract

This paper presents a functionality based model for survivable Intelligent Transportation System (ITS) control infrastructures. Specifically, a method is shown that allows us to determine the vulnerability of an ITS communication infrastructure to malicious acts and to derive optimal mitigations. The research is applied to a real ITS currently being implemented in a small town. Results show that significant reductions in vulnerability can be achieved utilizing the principle of redundancy with minimal effort. The method presented is especially suitable to reducing vulnerabilities of existing ITSs as it can be effectively applied after an ITS has been built. Mitigation techniques can be derived based on maximal benefit under consideration cost-benefit ratio.

1 Introduction

Over the last several years, attacks on computers and networks have reached epidemic proportions. Despite all efforts in the area of computer security, the number of reported incidents and vulnerabilities have roughly doubled each year [4]. In 2003 the Computer Emergency Response Team coordination center (CERT) logged over 137,000 incidents resulting from 3,784 reported vulnerabilities [4]. Yet we are fully embracing computers and networking technologies to control our nation's critical infrastructures including telecommunication, banking and finance, electrical power and transportation. The implications of malicious cyber acts with respect to critical infrastructures and their potential impact on national security have been pointed out in the 1997 Report of the President's Commission on Critical Infrastructure Protection [12]. To date, many questions about the risks involved remain unanswered [1, 6, 9].

Intelligent Transportation Systems (ITS), as envisioned by the Intermodal Surface Transportation Efficiency Act (ISTEA) of 1991, promise to improve the safety, efficiency, capacity and oversight capabilities of urban and rural roads, highways and thoroughfares [2, 10]. The specification and design of transportation systems, as with other critical infrastructures such as the electric power grid, have been based mainly on engineering principles addressing optimization and safety in relatively benign environments. However, ITS relies heavily on traditional computer networking and telecommunications technologies including fiber optic networks, microwave and wide area networks (WAN) along with components such as routers, switches, hubs and firewalls [2]. With increasing reliance on computers and communication networks, ITS have become as vulnerable to malicious acts as any networked computer system, raising great concerns about hacking and cyber terrorism and their potential consequences.

In the area of cyber terrorism, computer and network security and survivability are the principal research areas addressing protection. Security is often viewed as addressing issues of confidentiality, integrity and availability, as well as accountability and correctness. Survivability, on the other hand, addresses how to ensure that critical functionalities will function as specified even in the presence of malicious acts [5, 11]. Survivability goes beyond security and has been formulated in [5] with respect to *Resistance* to, *Recognition* of, and *Recovery* from attacks, with a final iteration considering *Adaptation*. Whereas resistance and recognition are typically associated with security, the main consideration of survivability is recovery. The recovery aspect can adopt many concepts from the area of fault-tolerance considering diverse fault models, which are directly affected by the topology and communication protocols of the systems involved [3, 7, 14].

This paper addresses vulnerability assessment and survivability of ITS with respect to reliability and attacks, both cyber and physical, on their control infrastructure. It introduces a model capable of aiding in the identification of mitigation techniques based on minimal application of the principle of redundancy, thus considering economic reali-

*This research was supported by a grant from the National Institute for Advanced Transportation Technology (NIATT)

†The author is on leave at the Laboratoire ID-IMAG, 38330 Montbonnot Saint-Martin, FRANCE

ties. Section 2 gives background information and describes the context of the research. Section 3 will give an overview of a new functionality based model. The model is applied to a real ITS and the analysis and results are presented in Section 4. Finally, Section 5 summarizes the paper.

2 Background

The methods and model presented have been developed within the context of a real ITS infrastructure that is currently being implemented in a rural town of population 25,000 through a joint effort between city, county and state agencies along with the National Institute for Advanced Transportation Technology (NIATT). This specific implementation, although small by comparison, is fundamentally representative of ITS implementations across the country. Furthermore, larger ITS implementations can be viewed as collections of smaller systems under consideration of complexity issues induced by scalability.

We are motivated by three factors: 1) The increased complexity and interconnectedness of ITS, 2) our ever-increasing reliance on automated traffic monitoring and control and 3) the identification of surface transportation as a Critical Infrastructure by the President's Commission on Critical Infrastructure Protection [12].

Given these, our goal is to develop an effective means of modeling the control infrastructure of ITS and their requisite functionalities with respect to vulnerability and survivability to cyber or physical attacks. The focus is on the ITS control infrastructure, applying the model directly to real-world ITS infrastructures for the purpose of enumerating critical elements, e.g. communications links, and suggesting a means of improving the reliability and reducing the vulnerability of ITS with respect to the critical elements.

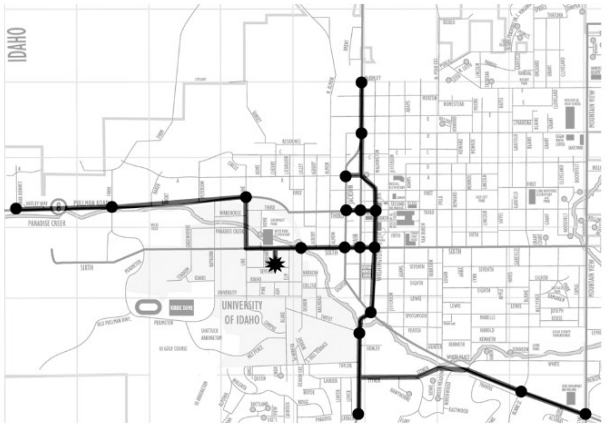

Figure 1: Street Map with ITS Infrastructure

The model introduced represents a static approach. This should not be confused with dynamic traffic modeling that would simulate the effects of the malicious act on the overall traffic flow. However, the static vulnerability and survivability analysis could/should be used as the basis for dynamic traffic modeling.

Figure 1 shows the municipal street map of the target ITS with the ITS control network superimposed. The dots represent regulated intersections and the asterisk represents a centralized traffic control center. Lines indicate the communication and network links.

The network in Figure 1 can be represented as a graph $G = (V, E)$, where the vertices $V_i \in V$ represent intersections (including signal heads, controllers, conflict monitors, etc.) and the edges $E_{i,j} \in E$ represent communication links. The resulting graph is shown in Figure 2. The centralized control center is represented by vertex V_0 in the graph; all communications to and from network components are routed through the graph to this single control center.

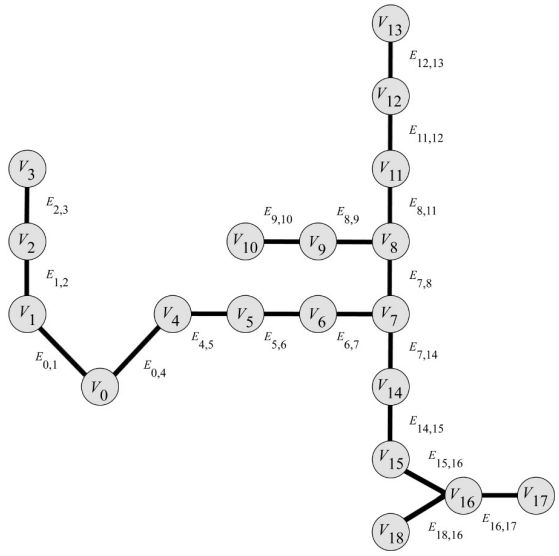

Figure 2: ITS Topology Graph

3 Model Overview

This section introduces a new ITS model, which, in order to reduce the complexity of an ITS infrastructure model, is viewed as a collection of functionalities. Specifically, it is assumed that the system is composed of, or capable of engaging, n functionalities F_i, where $1 \leq i \leq n$. A functionality is any usage scenario, e.g. updating a message board or changing the durations of a traffic light. A functionality F_i may in

turn contain other functionalities. If F_i cannot be composed of smaller functionalities, than it is said to be an *atomic functionality*. Each F_i affects certain physical components C_j. Examples of components include hubs, switches, message boards or closed circuit television (CCTV).

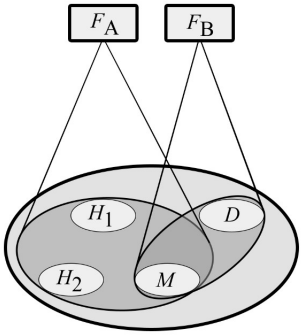

Figure 3: Mapping Model

Figure 3 illustrates the relationship between functionalities and components. If F_A denotes the functionality "changing the message of message board M from the control center" and M is reachable through two hubs H_1 and H_2, then the components associated with F_A are H_1, H_2 and M, and will be denoted by C_{H1}, C_{H2} and C_M. To keep the component notation intuitive during the examples below we will alternatively refer to a component C_x simply as x.

F_B, the second functionality in Figure 3, represents an alternative to F_A allowing "changing the message of message board M via access through dial-up modem D".

3.1 Set Mappings and Primitives

It will be useful to express a functionality in terms of the set of components it affects. Given F_i, let $\mathbf{C}_i$ be the set of its affected components. In the example of Figure 3, functionality F_A defines the component set

$$\mathbf{C}_A = \{H_1, H_2, M\}$$

and F_B defines

$$\mathbf{C}_B = \{D, M\}.$$

Component sets can be used to define functional primitives as will be described below.

Invulnerability Function: In order to capture the notion of the invulnerability of an ITS infrastructure, we introduce an invulnerability function. Given a set of components $\mathbf{C}$, let $\mathcal{V}(\mathbf{C})$ be a function that defines a quantitative measure, expressed as a probability, for the invulnerability of the components in $\mathbf{C}$ to a specific act. Such an act could be hacking or physical sabotage of the components. In terms of standard dependability considerations the "specific act" could be the "reliability" of the components in $\mathbf{C}$, e.g. considering aging of components. We intentionally consider invulnerability as it is analogous to reliability. Similarly, vulnerability is analogous to unreliability. This allows us to take advantage of reliability modeling used in Reliability Block Diagrams (RBD) and Fault Trees (FT) [13].

Series Invulnerability: If $\mathbf{C}$ consists of more than one component, and there are no redundant components, then $\mathcal{V}(\mathbf{C})$ is the product of the invulnerabilities of its components, i.e.

$$\mathcal{V}(\mathbf{C}) = \prod_{C_i \in \mathbf{C}} \mathcal{V}(C_i). \quad (1)$$

The definition is analogous to the definitions of the reliability of a series RBD [13], which is defined as the product of the reliabilities of the components of the series construct. This implies that, in the absence of redundancy, the reliability of a set of components is bound by the reliability of its least reliable component. With respect to vulnerability, the component group is at least as vulnerable as its most vulnerable component.

Parallel Invulnerability: If $\mathbf{C}$ consists of redundant components, i.e. $\mathbf{C}$ contains multiple components that implement the same functionality, one can model the component set as a parallel RBD. This assumes that the functionality represented by $\mathbf{C}$ can be performed as long as at least one component is still available. This is often referred to as a *1-of-N* configuration. It should be noted that this assumes a benign fault model, i.e. compromised functionalities simply crash, rather than performing incorrect actions.

Assume $\mathbf{C}$ consists of N components such that they implement a *1-of-N* configuration, then the resulting vulnerability is the product of the vulnerabilities of its components, i.e.

$$(1 - \mathcal{V}(\mathbf{C})) = \prod_{C_i \in \mathbf{C}} (1 - \mathcal{V}(C_i)). \quad (2)$$

This is analogous to the unreliability of a parallel RBD which is computed as the product of the unreliabilities of each component [13]. Note that since $\mathcal{V}(\mathbf{C})$ refers to an invulnerability, $1 - \mathcal{V}(\mathbf{C})$ refers to a vulnerability.

Selection Function and the *max* Primitive: Having several alternative, yet functionally equivalent, functionalities F_i allows for a choice based on a selection criteria. Recall that $\mathbf{C}_i$ is the component set of functionalities F_i. Given a numerical value $v = \mathcal{V}(\mathbf{C}_i)$, let $\mathcal{S}(v)$ be a selection function that maps v to a specific functionality F_i. This is use-

ful if one has a value v and wants to determine which functionality derived this value. Thus $\mathcal{S}(v)$ is an inverse function, i.e. if $\mathcal{C}(F_i)$ defines a function mapping F_i to $\mathbf{C}_i$, then $\mathcal{S} = \mathcal{C}^{-1} \circ \mathcal{V}^{-1}$. It should be pointed out that in order to simplify the discussion it is assumed that $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{V}$ are bijections, i.e. one-to-one and onto. If this were not the case, then $\mathcal{S}$ would be an inverse multi-function indicating sets of functionalities.

The selection function is useful if there are several alternative functionalities F_i. Given two functionalities F_i and F_j it is meaningful to select the functionality associated with the highest invulnerability, i.e. $v = \max(\mathcal{V}(\mathbf{C}_i), \mathcal{V}(\mathbf{C}_j))$. As a result, the least vulnerable alternate functionality is

$$\mathcal{S}(\max(\mathcal{V}(\mathbf{C}_i), \mathcal{V}(\mathbf{C}_j))). \tag{3}$$

In the example of Figure 3, the most secure choice to change the message of the message board is the functionality associated with the maximum invulnerability. Thus, if we compute

$$v_A = \mathcal{V}(\mathbf{C}_A) = \mathcal{V}(H_1)\mathcal{V}(H_2)\mathcal{V}(M)$$

and

$$v_B = \mathcal{V}(\mathbf{C}_B) = \mathcal{V}(D)\mathcal{V}(M)$$

then the least vulnerable alternative presents itself as

$$\mathcal{S}(\max(v_A, v_B)).$$

Again, since $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{V}$ are assumed to be bijections, there is always a unique F_i with highest invulnerability. If the bijection assumption does not hold then the result of the new $\mathcal{S}$ would be a set of functionalities F_i. From an invulnerability point of view, each F_i would be equivalent and thus any F_i could be selected.

Functional Primitive *composition*: Component sets can be expressed in terms of other component sets. Given two sets $\mathbf{C}_j$ and $\mathbf{C}_i$ such that $\mathbf{C}_j \subset \mathbf{C}_i$, then

$$\mathbf{C}_i = \mathbf{C}_j \bullet (\mathbf{C}_i \setminus \mathbf{C}_j) \tag{4}$$

where $\bullet$ is a composition operator, which in this case is simply the *set union* operator, and $\setminus$ indicates the *set minus* operator. The composition operator is useful when one wants to express functionalities or component sets in terms of other functionalities or component sets.

Let F_C be the functionality of accessing hub H_2 via H_1. The associated component set is $\mathbf{C}_C = \{H_1, H_2\}$. Assuming that F_A in Figure 3 and F_C both utilize the same communication path we can now express $\mathbf{C}_A$ in terms of $\mathbf{C}_C$, i.e.

$$\mathbf{C}_A = \{H_1, H_2, M\} = \mathbf{C}_C \bullet \{M\}.$$

Component set composition can be used to reduce computational complexities. Further, component set composition leads to functionality compositions. For example, the previous example can be extended to derive vulnerabilities, resulting in

$$\mathcal{V}(\mathbf{C}_A) = \mathcal{V}(\mathbf{C}_C) \bullet \mathcal{V}(\{M\}).$$

Now $\bullet$ is a composition operator that indicates parallel or series invulnerability utilizing equations (1) or (2).

4 Survivability Analysis

Our survivability considerations are the determination of the vulnerabilities of the ITS control infrastructure or individual functionalities as well as mitigation strategies reducing vulnerabilities. Survivability in the presence of loss of data or functionality requires redundancy [7, 8]. Whereas some survivability applications can utilize information redundancy or time redundancy, here spatial redundancy is required. For example, in the presence of loss of communication links or malicious tampering of ITS components, error codes or repeated execution of the compromised functionalities over time will not restore the functionality.

Adaptation of the ITS to disruptions or changes in traffic patterns can be part of a survivability strategy. For example, assume functionality F_D indicates "adapt control of the intersection to traffic volume observed by the CCTV". If one looses the CCTV or the controller update functionality due to a link disruption, then F_D fails. It is therefore important to design the ITS to incorporate alternate functionalities which constitute spatial redundancy. In the example of Figure 3, the alternate functionalities were represented by F_A and F_B, which implemented redundant functionalities to change the messages of message board M.

The ITS control configuration as shown in Figure 1 (abstracted in Figure 2) offers no alternative access functionalities and thus no redundancy. It represents a spanning tree, i.e. there are no cycles in the connected graph and, by definition, the connectivity of the tree is unity.

In the following analysis we will consider the ITS survivability based on functionality redundancy. It is assumed that an ITS infrastructure is already in place and that mitigation has to be optimized. No assumptions are made on the existing infrastructure, e.g. the ITS may have no survivability considerations with respect to link failure.

4.1 Analysis Assumptions

In order to demonstrate the model we will assume the ITS shown in Figure 1. The measure for vulnerability would usually consider parameters such as traffic volume, component visibility, communication protocols or proximity to critical assets such as hospitals, law enforcement or government. Therefore, invulnerability function $\mathcal{V}$ would be a multi-parameter function, implementing some multi-objective optimization strategy.

For the purpose of demonstration we will simplify the problem and only consider link failures as a function of link length. All other parameters are considered to be unity. This example may not be as limited as it seems, since link failures due to weather or physical disruption by construction equipment are not uncommon. Furthermore, we assume that the only functionality is access to components. Thus F_i is "accessibility of component V_i" in Figure 2.

Accessibility to components is dictated by the topology. Adding redundancy will be desirable, but will most likely be subject to cost constraints. Thus, the question arises as to where one should add a redundant link in order to maximize the induced overall invulnerability contribution. If one considers cost, then which additional link will bring the largest return on investment with respect to overall invulnerability gains?

Adding a single edge to the graph, which might be a spanning tree or may already contain cycles, will have affects on the invulnerability of the overall system it represents. Specifically, any edge inserted will produce a cycle. Such a cycle will result in *1-of-2* behavior, since components can be accessed from each direction of the cycle. In order to maximize overall invulnerability of the system the edge inserted must result in the maximum cumulative invulnerability, i.e.

$$\max(\sum_{i=1}^{n} \mathcal{V}(C_i)). \tag{5}$$

4.2 Analysis and Results

Considering only component accessibility based on link failures allows for invulnerability to be equated with link reliability $R(t)$. Let $G = (V, E)$ be the ITS graph, e.g. as in Figure 2, with vertex set V and edge set E. Then for each edge $E_{i,j} \in E$ defined by vertices $V_i, V_j \in V$ one can define the link invulnerability, i.e. reliability, as

$$\mathcal{V}(E_{i,j}) = R(t) = e^{-w\lambda t}$$

where w is the link weight of $E_{i,j}$, λ is the link fail rate, and t defines the time interval under consideration. $R(t)$ thus is the standard definition of reliability, i.e. the probability that the system is functional throughout the entire time interval [0,t]. Table 1 shows the weights of each edge in the ITS graph. The weights are the actual distances measured in feet and are not hypothetical values. They represent the length of the edges in Figure 2.

Based on G, one can derive a Reliability Block Diagram. The RBD is the series-parallel representation of G, with edge $E_{i,j}$ representing a block with associated fail rates induced by edge weight $w_{i,j}$ and λ, i.e. $w_{i,j}\lambda$.

The analysis for single edge insertion was based on cumulative invulnerabilities, i.e. $\sum_{i=1}^{n} \mathcal{V}(C_i)$, with the best edge giving the maxima shown in Equation (5). The functionalities F_i were defined as accessibility of V_i from the traffic center V_0. Figure 4 shows the impact of single edge insertion on the overall cumulative reliabilities. We use the term reliabilities rather than invulnerabilities to remind the reader that the fail rates were determined relative to the link length. For each edge two values are shown, computed for a fixed $\lambda t = 0.001$. The gray bars indicate the cumulative reliability gains (in percentage) resulting from adding the edge specified. Since adding an edge can be expensive, the reliability gains were normalized by the edge length, i.e. link cost. The resulting cost-benefit ratios for each edge are shown as black bars.

The results for cumulative reliabilities are shown for the 20 best edges. As can be seen, the highest gain was derived by inserting edge $E_{1,12}$ into the graph. Its insertion alone increased the cumulative reliability of the ITS by 23%.

The cost-benefit ratios for different edges vary greatly. For example, edges $E_{2,11}$ and $E_{5,10}$ show roughly the same

$E_{0,1}$	2000	$E_{9,10}$	800
$E_{1,2}$	3600	$E_{8,11}$	1500
$E_{2,3}$	4300	$E_{11,12}$	1300
$E_{0,4}$	1000	$E_{12,13}$	1300
$E_{4,5}$	1200	$E_{7,14}$	2000
$E_{5,6}$	800	$E_{14,15}$	800
$E_{6,7}$	800	$E_{15,16}$	5000
$E_{7,8}$	1200	$E_{16,17}$	4300
$E_{8,9}$	800	$E_{16,18}$	5000

Table 1: Edge Weights of ITS Graph

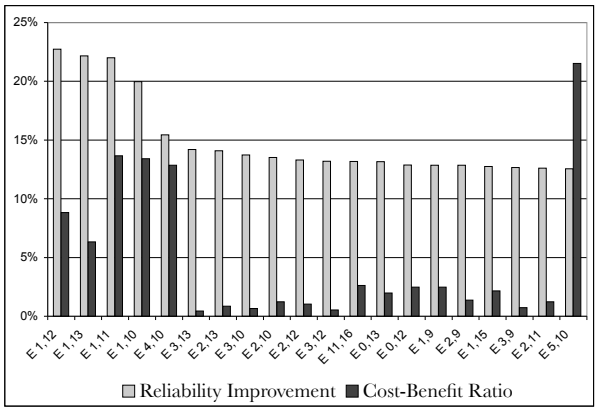

Figure 4: Impact of Redundancy on Invulnerability

cumulative reliability improvements of 13% but their cost-benefit ratios differ greatly. As a result, out of the two edges, $E_{5,10}$ is the far better choice. Considering cumulative reliability improvements in conjunction with cost-benefit ratios allows system designers to make intelligent decisions about which links to add to the ITS. With respect to the ITS, the most sensible choice is $E_{1,11}$, resulting in reliability improvements of 22% at 2/3 and 1/2 the cost of the best and second-best choice.

It should be noted that the sizable improvements observed were achieved by a single edge. Iterative repeated application of the method, or inclusion of multiple edges in one step, can derive far greater invulnerabilities.

As expected, invulnerabilities varied over different λt. Many interesting conclusions could be drawn from the results, but due to space restrictions they could not be displayed and discussed here.

5 Summary

This research presented a view of an ITS control infrastructure based on a hierarchical model of functionalities. Each functionality defined component sets which were the basis for determining its invulnerability to malicious acts. Series and parallel invulnerabilities were defined to allow the computation of invulnerabilities of series-parallel component configurations. These computations were the basis for identifying mitigations minimizing overall vulnerability.

The method was applied to a real-world ITS control infrastructure considering communication link failures. It was shown that by inducing link redundancy, vulnerability could be significantly reduced. The selection of the redundant links was based on maximizing invulnerability. Cost-benefit ratios allowed for selecting links in a cost effective manner. This addressed the economic considerations of ITS operators since the inclusion of links, especially in a post implementation phase, can be costly.

References

[1] M. Amin, "Toward Self-Healing Infrastructure Systems", *IEEE Computer*, Vol. 33(8), pp. 44-53, August 2000.

[2] "Application of Advanced Transportation Technology Within Washington State: Discussion and Policy Recommendations", The Committee for Advanced Technology in State Transportation Policy, June, 1999. Available at: http://citeseer.nj.nec.com/ps/314534

[3] O. Babaoglu, and R. Drummond, "Streets of Byzantium: Network Architectures for Fast Reliable Broadcasts", *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 6, pp. 546-554, June 1985.

[4] CERT Coordination Center, CERT/CC Statistics 1988-2003, Software Engineering Institute, Carnegie Mellon, http://www.cert.org/stats/cert_stats.html.

[5] E. Ellison, L. Linger, and M. Longstaff, "Survivable Network Systems: An Emerging Discipline", Carnegie Mellon, SEI, Technical Report CMU/SEI-97-TR-013, 1997.

[6] A. Jones, "The Challenge of Building Survivable Information-Intensive Systems", *IEEE Computer*, Vol. 33(8), pp. 39-43, August 2000.

[7] A. W. Krings, W.S. Harrison, A. Azadmanesh, and M. McQueen, "Scheduling Issues in Survivability Applications using Hybrid Fault Models", to appear in Parallel Processing Letters, also available at http://www.cs.uidaho.edu/ krings/publications.html.

[8] A. W. Krings, "Agent Survivability: An Application for Strong and Weak Chain Constrained Scheduling", 37^{th} *Hawaii International Conference on System Sciences*, (HICSS-37), Minitrack on Security and Survivability in Mobile Agent Based Distributed Systems, paper STSSM01, 8 pages, January, 2004.

[9] T. Longstaff, C. Chittister, R. Pethia, and Y. Haimes, "Are We Forgetting the Risks of Information Technology?", *IEEE Computer*, Vol. 33(12), pp. 43-51, December 2000.

[10] "The National ITS Program: Where We've Been & Where We're Going", U.S. Department of Transportation, Report No. FHWA-JPO-97-0027, March, 1997.

[11] P. Neumann, *Practical Architectures for Survivable Systems and Networks*, (Phase-Two Final Report), Computer Science Laboratory, SRI International, June 2000.

[12] The Report of the President's Commission on Critical Infrastructure Protection, Critical Infrastructure Assurance Office, Washington, D.C. Oct. 1997. Available: http://www.ciao.gov/resource/pccip/PCCIP_Report.pdf

[13] R. Sahner, K. Trivedi, and A. Puliafito, "Performance and Reliability Analysis of Computer Systems, An Example-Based Approach Using the SHARPE Software Package", Kluwer Academic Publishers, 1996.

[14] P. Thambidurai, and You-Keun Park, "Interactive Consistency with Multiple Failure Modes", *7th Reliable Distributed Systems Symposium*, Columbus, OH, pp. 93-100, October 1988.