On the Definition of Survivability

John C. Knight* Kevin J. Sullivan
Department of Computer Science
University of Virginia
151 Engineer's Way
P.O. Box 400740
Charlottesville, VA 22904-4740
(804) 924 2200 (Voice)
(804) 982-2214 (FAX)
{*knight* | *sullivan*}*@cs.virginia.edu*

Abstract

The computer systems that provide the information underpinnings for critical infrastructure applications, such as banking systems and telecommunications networks, have become essential to the operation of those applications. Failure of the information systems will often cause a major loss of service, and so their dependability has become a major concern. Current facets of dependability, such as reliability and availability, do not address the needs of critical information systems because they do not include the notion of degraded service as an explicit requirement. What is needed is a precise notion of what forms of degraded service are acceptable to the application, under what circumstances each form is most useful, and the fraction of time degraded service levels are acceptable. This concept is termed survivability and in this paper we show that it is a necessary new facet of dependability. We present a formal definition of survivability and present an example of its use.

Keywords

Survivability, critical information systems.

Submission Category

Regular paper.

Declaration

This paper has been cleared through the authors' affiliation.

1. Introduction

Many large infrastructure systems have evolved to a point where organizations rely heavily upon them. In some cases, such systems are so widespread and so important that the normal activities of society depend upon their continued operation; examples in this latter category include transportation systems, telecommunications networks, energy generation and distribution systems, and financial services. Such systems are usually referred to as *critical infrastructure applications.*

Powerful information systems have been introduced into critical infrastructure applications as the cost of computing hardware has dropped and the availability of sophisticated software has increased [6]. In many cases, the provision of service by infrastructure applications is now highly dependent on the correct operation of computerized information systems, and, frequently, damage to the information system will lead to a loss of at least part of the service provided by the infrastructure application. In some cases, relatively minor damage can lead to a complete cessation of service. We refer to such information systems as *critical information systems*.

The dependability of these critical information systems has become a major concern [12, 13]. Dependability is a system property that is usually stated as a set of requirements with which the system has to comply. Dependability has many facets—reliability, availability, safety, and so on [7]—and to permit exact requirements statements about systems, each of these terms has a precise meaning. For example, the reliability of a system, R(t), is defined to be the probability that the system will meet its requirements up until time t when operating in a prescribed environment. Similarly, the availability of a system, A(t), is the probability that the system will be operating correctly at time t. For systems for which dependability is important, the system requirements state the minimum acceptable value for the relevant facet of dependability, such as R(t) or A(t), and it is then the responsibility of the developers and operators to show that the necessary dependability will be achieved during system operation.

Different facets of dependability are suitable for different systems—highly reliable operation is usually needed for an embedded control system, highly available operation is usually needed in a database system, and a high level of safety is needed for a weapons system. It is important to note that a system might achieve one facet of dependability but not others. For example, a system that fails frequently but only for very brief periods has high availability but low reliability. Many systems are built to operate this way intentionally because it is a cost-effective approach to providing service if reliability (in the formal sense) is not required but availability is.

In specifying dependability for a given system, it is usually the case that full system functionality is required—nothing is usually stated beyond perhaps failure semantics. For critical infrastructure applications, this is insufficient. Some events that damage a system have no external effect because of appropriate redundancy; mirrored disks for example. But in other cases, damage is so widespread that functionality has to be changed. A wide-area loss of commercial power, for example, might force a critical on-line database service to switch to a remote back-up site that has less throughput capacity or reduced functionality. Such circumstances arise with sufficient frequency in infrastructure applications that comprehensive provision for them must be made. Thus, the service changes that a system might be forced to provide during routine operation must be specified, and users of the system need to be aware that changed service is a possibility.

The prospect of damage that forces a change in service, combined with several others characteristic that we present in section 3, leads to the conclusion that the existing facets of dependability—reliability, availability, and so on—are not sufficient to capture the essential dependability demands for critical information systems. In this paper we review and define formally a relatively new facet of dependability—*survivability.* A precise definition of survivability is important if we are to build survivable systems. If we do not state accurately what we mean by a system being survivable, we cannot determine whether we have made a system that is survivable.

The remainder of this paper is organized as follows. In the next section we review related work and in section 3 we present a brief summary of two critical infrastructure applications to provide the context for survivability. In section 4 we discuss the intuitive notion of survivability and our formal definition. To illustrate the various aspects of the definition, we present an example in section 5. In sections 6 and 7 we discuss the relationship between survivability, fault tolerance, and security. Finally, in section 7, we present our conclusions.

2. Related Work

The notion of survivability has been used in several engineering disciplines outside of critical information systems. For example, it is a common concept in weapons systems engineering [10, 3]. The survivability of combat aircraft has emerged as a subject of intense study, and a definition has been created for aircraft combat survivability [18]:

> **Survivability:** Aircraft combat survivability is the capability of an aircraft to avoid and/or withstand a man-made hostile environment. It can be measured by the probability the aircraft survives an encounter with the environment, P_S.

The Institute for Telecommunications Services, a part of the U.S. Department of Commerce, has created an extensive glossary of telecommunications terms in Federal Standard 1037C [16]. This glossary contains a definition of survivability for telecommunications systems:

> **Survivability:** A property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance; e.g., nuclear burst. Note: For a given application, survivability must be qualified by specifying the range of conditions over which the entity will survive, the minimum acceptable level or post-disturbance functionality, and the maximum acceptable outage duration.

Both of these definitions are seeking a framework to define service after some form of damage, and they relate closely to our goal of defining survivability for critical information systems. It is also interesting to note that both definitions are probabilistic. Finally, we note that the second definition includes the notion of degraded or different service, and requires that it be defined.

In the context of software engineering, Deutsch has offered the following definition [4]:

> **Survivability:** The degree to which essential functions are still available even though some part of the system is down.

This definition is not sufficient for our needs. If it were applied to a critical information system in this form, the user of the system could not be sure which functions had been selected as “essential functions” nor under what circumstances (i.e., after what damage) these functions would be provided.

In earlier work specifically on information system survivability, Ellison et al. introduced the following definition [5]:

> **Survivability:** Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures, and recover full services in a timely manner

While this definition is a good beginning, again it does not have the precision needed to permit a clear determination of whether a given system should be considered to be survivable. The first problem is that much is implied by the phrases “essential services”, “attacks and failures”, and “timely manner”. If nothing further is defined, it is not possible for the developer of a system to determine whether a specific design is adequate to meet the needs of the user community. More importantly, if a phrase such as “essential service” is not precisely defined, it might be the case for any specific system that the determination of what constitutes an essential service is left to the system’s developers rather than being defined carefully by application experts.

A second problem with a definition of this form is that it provides no testable criterion for the term being defined. By contrast, the definition of reliability makes a clear distinction between our *informal* view of a system as being reliable (it “never” fails) and the *formal* view provided by the definition which is that a system is reliable if it meets or exceeds a probabilistic goal. By that definition, a system might fail and yet still be formally considered reliable. The same degree of clarity is needed for survivability so that we may consider a system to be survivable and know what that means.

In the field of information system survivability more generally, a body of research results has begun to appear. A valuable source of material is the series of Information Survivability Workshops [17]. Many relevant papers have appeared in various other conferences concerned with dependability.

The concept of *performability* is related in a limited way to survivability [8]. A performability measure quantifies how well a system maintains parameters such as throughput and response time in the presence of faults over a specified period of time [9]. Thus performability is concerned with analytic models of throughput, response time, latency, etc. that incorporate both normal operation and operation in the presence of faults. As we show later in this paper, survivability is concerned primarily with system functionality, and precise statements of what that functionality should be in the presence of faults.

3. Critical Infrastructure Applications

Some background material about critical infrastructure applications is helpful in understanding the need for a precise notion of survivability and how it differs from other facets of dependability. Detailed descriptions of four applications are available elsewhere [6]. In this section we summarize two applications very briefly, and then outline a set of important characteristics that tend to be present in critical information systems. These characteristics are, in some cases, unique to this class of system, and they affect the way that dependability is both perceived by the user and addressed by the designer.

3.1. Banking and Financial Services

The nation's banking and finance systems provide a very wide range of services—check clearing, ATM service, credit and debit card processing, securities and commodities markets, electronic funds transfers, foreign currency transfers, and so on. These services are implemented by complex, interconnected, networked information systems.

The most fundamental financial service is the financial payment system [15]. The payment system is the mechanism by which value is transferred from one account to another. Transfers might be for relatively small amounts, as occur with personal checks, all the way up to very large amounts that are typical of commercial transactions. For a variety of practical and legal reasons, individual banks do not communicate directly with each other to transfer funds. Rather, most funds are transferred in large blocks by either the Federal Reserve Bank or by an Automated Clearing House (ACH).

The basic functionality of the payment system leads to a system architecture in which there are tens of thousands of leaf nodes in the network that provide customer access. These nodes are connected in various ways to a much smaller number of intermediate nodes that provide regional service or centralized service associated with a specific commercial bank. Finally, those nodes are connected to a few nodes that provide communication for value transfer between separate commercial entities. This architecture is illustrated in Figure 1.

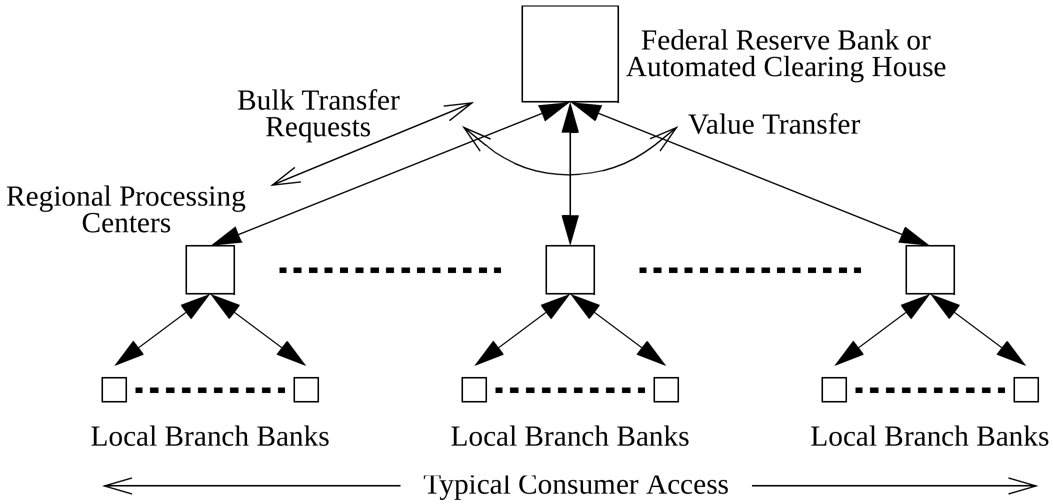

Figure 1. Hypothetical banking network

To illustrate application-domain characteristics in more depth, we use the actions required in clearing a check as an example. Check clearing is a complicated procedure and we present a brief summary of *only one* of the ways in which checks are cleared—there are several variations.

The recipient of a check presents the check for collection at a branch of the bank holding his or her account (i.e., a network leaf node). The check is scanned, the essential detail recorded (amount, payee, etc.) in electronic form, and then the paper check is often destroyed. The electronic form of the check is then forwarded to a regional processing center (i.e., a network intermediate node) where details are recorded in a database. The check is then forwarded to a clearing house or the Federal Reserve Bank (i.e., a root node) where accumulation data describing what one retail bank owes or is owed by other retail banks is computed. Periodically (usually once per day) this data is used to move funds between master accounts that the retail banks maintain. Finally, transactions flow back down through the system eventually authorizing the recording of the deposit to one customer's account and a debit from another (although the funds will already have moved). High-value commercial funds transfers are almost always purely electronic transfers and they are handled individually and upon presentation rather than at some scheduled time as part of a block.

3.2. Freight Rail Transportation

The freight-rail transport system moves large amounts of raw materials, manufactured goods, fuels, and food [2]. Operation of the freight-rail system uses computers extensively for a variety of purposes. For example, *every* freight car in North America is tracked electronically as it moves and very large databases are maintained of car and locomotive locations. Tracking is achieved using transponders on the freight cars and track-side equipment that communicates in real time with computers maintaining the databases. This data permits tracking of specific shipments, sometimes by allowing customer computer systems to have access to rail-system computers. The data also permits scheduling of freight cars for individual trains—a massive task given that there are hundreds of thousands of freight cars that might be anywhere within the rail system at any point. Knowledge of what trains are where, where they are going, and the capacity of the rail infrastructure also permits overall rail-network management. Thus, which trains use which track can be managed so as to improve factors such as on-time arrival rate, throughput, and so on.

At present, actual train movement is under human control for the most part although controlling locations are very few in number and frequently hundreds of miles from controlled locations. An especially important application that is being used increasingly in the freight-rail system is *just-in-time* delivery. Train movements are scheduled so that, for example, raw materials arrive at a manufacturing plant just as they are required. This type of service necessitates analysis of demands and resources over a wide area, often nationally, if scheduling choices are to be made that permit trains to arrive at the required time.

3.3. Critical Information System Characteristics

Society now faces an unquantified but apparently serious risks that reliance on fragile and insecure information systems will compromise delivery of critical civic infrastructure services. Massive computerization has enabled efficiencies through tightened coupling. Just-in-time delivery of automotive parts by rail, for example, has enabled dramatic inventory reductions; but manufacturers are now more reliant on a highly reliable stream of timely deliveries. The cost of disruptions grows more rapidly in time now than before computerization, yet increasing reliance on computers increases vulnerability to disruption. The central problem that we face is to devise approaches to infrastructure information systems design and evolution that simultaneously enable the efficiencies that computers make possible while ensuring that the costs of service stream interruptions remain acceptable in the face of disruptions to underlying information systems.

The scale, sophistication, and makeup of infrastructure applications, such as those summarized earlier in this section, complicate the notion of dependability considerably. It is not useful to speak of the reliability of the freight-rail system, for example, because it is sure to be the case that parts of it will be non-operational at any time with little effect. However, there are some forms of damage that would be extremely serious. A total loss of the data recording freight-car locations would cripple the entire system because equipment could not be located. A partial loss of data recording or a large increase in access times would disrupt train movements, possibly over a wide area. Flooding or other environmental damage could impede planned movements and disrupt just-in-time service.

Events that disrupt critical infrastructure applications are inevitable and must be dealt with in some way. The continued provision of some form of service is more than desirable—in many cases it is essential. In the financial payment system, for example, if full service cannot be maintained, then it might be imperative that large commercial and governmental funds transfers be completed.

For the developer of a critical information system, knowing what service is required in the event that full service cannot be provided is very important. This information is essential input to the design process for the critical information system since achieving even some form of reduced service will almost certainly necessitate specific design choices. A notion of dependability for critical infrastructure applications and thereby for critical information systems is needed, but the current facets (reliability, availability, etc.) do not provide the necessary concepts. The problem lies in the fact that they do not include the notion of degraded service and the associated spectrum of factors that affect the choice of degraded service as an explicit requirement. The term that has come into use for this new facet of dependability is survivability.

To provide a basis for a discussion and to guide a definition, we enumerate the various characteristics of infrastructure applications that affect the notion of survivability. The characteristics are:

- *System Size* Critical information systems are *very* large, both geographically and in terms of numbers and complexity of computing and network elements. It is infeasible to engineer such systems so that none of their components fail during normal periods of operation yet scale precludes comprehensive redundancy.
- *Externally Observable Damage* In some cases, the effects of damage to a system will be so extensive that it will be visible to the system's users in the form of a change in service or the quality of service. Externally observable damage must be both expected and dealt with.
- *Damage and Repair Sequences* Events that damage a system are not necessarily independent nor are they necessarily mutually exclusive. In practice, a sequence of events might occur over time in which each event causes more damage—in effect, a bad situation gets progressively worse. It is likely, therefore, that a critical infrastructure application will experience damage while it is in an already damaged state, and that a sequence of partial repairs might be conducted. Thus, a series of changes in functionality might be experienced by a user with progressively less service available over time as damage increases and progressively more available

as repairs are conducted. This might happen, for example, if a computer network is subjected to a cascading failure in which the failure of one node leads to the failure of others and so on. A major source of increasing damage is likely to be coordinated security attacks in which a series of deliberate, malicious acts are effected by a group of attackers over some period of time.

- *Time-Dependent Damage Effects* The impact or loss associated with damage tends to increase with time. The loss associated with brief (seconds or less) interruptions of electric power can be mitigated in many cases. A protracted loss (days) is much more serious with impact tending to increase monotonically with time. Similarly, a protracted failure of freight-rail transportation would be devastating. The impact would not be immediate because freight is carried in bulk, but the effect would be noticeable within days. And it would become increasingly serious as time passed
- *Heterogeneous Criticality* The requirements for dependability in infrastructure systems are considerable as would be expected, but requirements vary with function and with time. It is important in power generation, for example, to maintain power supply if possible, but it is not necessary to maintain an optimal generation profile. In addition, some customers, such as manufacturing, will tolerate occasional lengthy interruptions with a sufficiently large rate reduction incentive whereas passenger transport systems typically cannot tolerate power interruptions. Criticality varies among infrastructures, among customers, and even over time. Longer-term power outages are more critical to hospitals than to homes, and more critical in winter than in summer. Finally, freight rail service is especially critical in October—harvest time.
- *Complex Operational Environments* The operating environments of critical infrastructures are of unprecedented complexity. They carry risks of natural, accidental, and malicious disruptions from a wide variety of sources; sometimes highly variable loads that vary both over time and space; varying levels of criticality of service; and so forth. Moreover, operational environments now exhibit previously unrealized behaviors such as widespread, coordinated information attacks.
- *Time-Varying Operational Environments* The operating environments of critical infrastructure applications are changing with time. For example, Internet access to freight-rail cargo records is available now to customers but was not five years ago. Similarly. security threats have increased dramatically from negligible levels to significant threats in recent times. For a sys-

tem to be viewed as survivable for any protracted period of time, the possibility of changes in the environment over time must be considered.

The factors in this list combine to present a picture of critical information systems that is quite different from computer systems that exemplify traditional dependability requirements. Avionics systems which typically require high reliability and telecommunications switches which typically require high availability possess few of the characteristics listed above. Dealing with all of these issues is essential if a particular critical information system is to be viewed as survivable. With this in mind, we proceed to formulate a definition of survivability.

4. Survivability

4.1. The Intuitive Notion

As we have discussed in an earlier paper [14], an infrastructure provides a service stream to its customers over time. For example, the electric grid provides a stream of electricity to each home and business. Such a stream has a *value* or worth to a customer depending on the customer's particular needs. At some level, there is an aggregate value added that depends on the reliance of customers on service and the criticality of each customer in a broader context, as defined by societal or business policy.

The direct consequence of a disruption of an infrastructure system is a reduction in the service stream to customers. The key consequence is a loss of value over time. The value lost to a given customer depends on the customer's reliance on the service, and the characteristics of the disruption. Individual loss sums to an aggregate loss, and an important notion for service providers is to minimize this aggregate loss such that the loss is judged acceptable under defined adverse circumstances.

Informally by a *survivable* system we mean a system that has the ability to continue to provide service (possibly degraded or different) in a given operating environment when various events cause major damage to the system or its operating environment. Service quality, i.e., exactly what a system should do when damaged, seems to be something that should be determined by simple guidelines. Intuitively, it would seem that the more functionality that could be provided the better, and that the more of the customers' needs that are met the better. It is certain to be the case that customers expect the "usual" functionality "most" of the time, and, depending on the type of damage, different subsets of "critical" functionality if the system has been damaged.

In practice, any intuitive notion of service quality, including this one, is not sufficient for systems of this complexity. In fact, the appropriate goal of survivability is to maintain as much of the fundamental customer value of the service stream as is cost-effective. Given the characteristics of critical information systems outlined in the previous section, maintaining value is a very difficult prospect but the concept of survivability has to capture this notion.

Refining the informal notion of survivability somewhat, we observe that survivability needs to specify the various different forms of tolerable service that the system is to provide given the notion that circumstances might force a change in service to the user. A tolerable (but not necessarily preferred) service is one that combines functions which work harmoniously and which will provide value to the user. The set of tolerable services are the different forms of service that the system must be capable of providing. At any one time, just one member of the set would actually be operating—the others represent the changed or degraded service definitions.

Along with the set of tolerable forms of service, a statement is required about which of them is preferred and under what circumstances. This is the way in which quality of service has to be addressed. The value that each of the various forms of tolerable service provides to the user must be documented, and by ordering these values, it is immediately clear what the service priority is. In addition, since the integral of value over time is a key metric, a probability has to be defined for each form of tolerable service that defines what fraction of the time that service must be provided on average. We note that the value associated with a specific tolerable service is a complex function of a several variables possibly including the users' state, calendar time, and so on.

This notion of survivability of computing systems is not new in that many critical systems have requirements for reduced or alternate service under some circumstances. The reason for making survivability a new facet of dependability is that it is a primary form of dependability needed by critical networked information systems. By defining it precisely, system owners can state exactly what the user of a system can expect over time in terms of the provision of service, and system designers have a precise statement of the dependability that the system has to achieve and can design accordingly.

4.2. Defining Survivability

All existing facets of dependability are defined in terms of a specification. For example, if a reliable system is needed, we state what we want by requiring that the system meet the functionality requirements with a certain probability assuming a certain environment. Our approach to defining survivability, therefore, starts with a specification statement:

Survivable System: A system is survivable if it complies with its survivability specification.

Building on the informal notion of survivability introduced above, we define a survivability specification precisely using the following definition:

Survivability Specification: A survivability specification is a four-tuple, {E, R, P, M} where:

E = A statement of the assumed operating environment for the system.

R = A set of specifications each of which is a complete statement of a tolerable form of service that the system must provide.

P = A probability distribution across the set of specifications, R.

M = A finite-state machine denoted by the four-tuple {S, s_0, V, T} with the following meanings:

S: A finite set of states each of which has a unique label which is one of the specifications defined in R.

s_0: $s_0 \in S$ is the initial or preferred state for the machine.

V: A finite set of customer values.

T: A state transition matrix.

The meaning of this four-tuple is as follows:

- *Environment*—E

  E is a definition of the environment in which the survivable system has to operate. It includes details of the various hazards to which the system might be exposed together with all of the external operating parameters. To the extent possible, it must include any anticipated changes that might occur in the environment.

- *Specification Set*—R

  R is the set of specifications of tolerable forms of service for the system. This set will include one distinguished element that is the normal or *preferred* specification, i.e., the specification that provides the greatest value to the user and with which the system is expected to comply most of the time. It is worth noting that at the other extreme, a completely inert system, i.e., no functionality at all, might be a tolerable member of this set.

- *Probability Distribution*—P

  A probability is associated with each member of the set R with the sum of these probabilities being one. The probability associated with the preferred specification defines the fraction of operating time during which the preferred specification must be operational. It is a lower bound, and will be close to one since the service defined by that specification must be present most of the time. The probabilities associated with the other specifications are upper bounds and define the maximum fractions of operating time that the associated specifications can be operational.

This probability distribution is designed to set a quantifiable limit on performance and is used to permit appropriate design decisions to be made. If, for example, hazard and other forms of analysis reveals that the probability associated with the preferred specification cannot be met with a specific design, then either the probability has to be changed or the system redesigned.

A severe practical limitation is brought about at this point by our inability to predict certain quantities that will be needed in showing analytically that the stipulated probabilities are met. The critical quantities that cannot be determined are the probability of failure of most forms of software and the probability of a malicious attack against a system. In order to allow analysis to proceed at least somewhat, the members of the set P have to be treated as *conditional* probabilities that are conditional on some assumed level of software failure and malicious attack.

- *Finite-state Machine*—F

  F defines precisely how and when the system is required to move from providing one form of tolerable service to another. Each state in S is associated with one of the specifications of tolerable service in R, and one state (s_0) will be labelled by the preferred specification.

  Defined transitions from one state to another (T) document the transitions between different forms of tolerable service that are possible. Some transitions will be from higher value states to lower value states, and they will be taken when the higher value state can no longer be maintained following damage. Similarly, some transitions will be from lower value states to higher value states, and they will be taken when the higher value state can be resumed following repair. The computation of customer value associated with the different states (V) has to be an on-going activity since value changes with time and other factors.

5. An Example

To illustrate the definition, we present an example based on a hypothetical financial payment system. We assume the network topology shown in Figure 1 in which there are a large number of nodes associated with “branch” banks (small retail institutions), a smaller number of “money-center” banks that are the primary operations centers for major retail banking companies, and a small set of nodes that represent the Federal Reserve Bank.

We summarize the environment for the example by assuming that all of the expected elements are defined and that the identified hazards include: major hardware disruption in which communi-

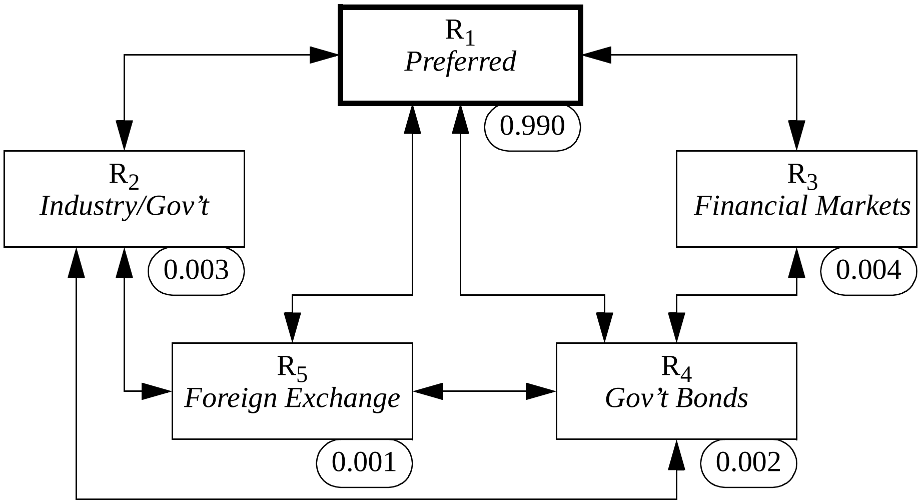

Figure 2. Survivability specification example.

cations or server machines become non-operational; coordinated security attacks in which multiple commercial bank regional processing centers are penetrated; and regional power failure in which either many branch banks are disabled or several money-center banks are disabled.

For this example, we assume several forms of tolerable service for the payment system:

- R$_1$ *Preferred.* This specification defines complete and normal functionality. Services that are included are electronic funds transfers, check processing, support for financial markets such as stocks and bonds, and international funds transfers.
- R$_2$ *Industry/Government.* This specification limits service to major industrial and government clients only. Services are restricted to electronic transfer of large sums.
- R$_3$ *Financial Markets.* This specification defines service for all the major financial markets including the stock, bond and commodity markets, but no other client organizations.
- R$_4$ *Government Bonds.* This specification defines service for processing of sales and redemptions of government bonds only and only by major corporate clients.
- R$_5$ *Foreign Transfers.* This specification defines service in which transfers of foreign currency into or out of the country are the only available service.

Figure 2 shows the various transitions between tolerable forms of service. Below each state in the figure is the probability for that state from the set P.

To decide upon the customer values (i.e., the value seen by users of the payment system) associated with the states in this example, we note: (1) that settlement by the clearing houses and the

Federal Reserve Bank occurs during the late afternoon; (2) domestic markets are closed at night; and (3) stock, bond, and commodity markets must be accommodated when trading volumes are exceptionally and unexpectedly high. Clearly, customer values associated with the financial payment system vary dramatically over time. There is little value to providing processing service to domestic financial markets over night, for example, and thus international transfers of funds have higher value. Similarly, extreme volume in domestic financial markets leads to a high demand on the financial payment system and this demand must be met if possible. Thus during periods of high volume, service to financial markets has very high customer value, almost certainly exceeding, for example, that for international funds transfers.

During times of political crisis, sentiment turns away from most traditional financial instruments and government bonds become heavily sought after. Thus, during such times, the ability to maintain a market in government bonds is crucial. Most other services might not be. To see an example of the impact of crisis on the financial system, see the failure that occurred in November 1985 when a 16-bit counter overflowed in the software operated by the Bank of New York in its government bond trading system [11].

In our example, consider first the occurrence of a fault with a major server that occurs during the middle of a normal market day and which cannot be masked. The options that the system has are to move to either state R_2, R_5, or R_3, (see Figure 2) and the maximum value (hypothetically) would be in state R_2 in this case. Were this to occur during the night, the transition would be to state R_5. Now suppose that while the server is down, a coordinated security attack is launched against the system (a bad situation getting worse). In that case the system would move to state R_4 since that would permit the best support in the event that the situation developed into a governmental crisis.

If a regional power failure were to occur that affected hundreds of local branch banks, there would probably be no change in the system state because there would be no impediment to operating the preferred specification. That large numbers of customers would be inconvenienced is unfortunate but not a problem for the information system. If however a large number of local branch banks reported intrusion alarms being triggered then a very different situation exists. In that case, to protect the network ability to serve its customers, it would make sense to transition to state R_2 in order to restrict access and protect the most valuable clients.

6. Survivability and Fault Tolerance

6.1. The Role of Fault Tolerance

The informal notion of an event that causes damage which we have used is referred to formally as a *fault* [1]. In many cases, systems are built using techniques of replication so that the effects of a fault do not affect the system's external behavior. Such faults are said to be *masked*. Usually for economic or similar practical reasons, some faults are *non-masked*; that is, their effects are so extensive that normal system service cannot be continued with the resources that remain even if the system includes extensive redundancy. These concepts of masked and non-masked faults are the formal statements of the idea introduced above of events that cause damage whose effects cannot or can be observed in the system's behavior.

Survivability is a dependability property, it is not synonymous with fault tolerance. Fault tolerance is a mechanism that can be used to achieve certain dependability properties. In terms of dependability, it makes sense to refer to a system as reliable, available, secure, safe, survivable and so on, or some combination using the appropriate formal definition(s) [7]. Describing a system as fault tolerant is really a statement about the system's design, not its dependability.

While fault tolerance is a mechanism by which some facets of dependability might be achieved, it is not the only mechanism. Other techniques, such as fault avoidance, can be used also. Thus, for example, by careful component selection it might be possible to reduce the rate of hardware failures in a given system to a negligible level, and by suitably restricting system access it might be possible to eliminate certain types of security attacks. In similar ways, fault elimination and fault forecasting can be used as mechanisms to improve a system's dependability.

6.2. Implementing Survivability

Survivability is a system property that can be required in exactly the same way that the other facets of dependability can be required. There is no presumption about how survivability will be achieved in the notion of survivability itself—that is a system design and assessment issue. However, the probabilities associated with each of the tolerable forms of service are important design constraints since they will determine which design choices are adequate and which are not.

A practical survivability specification will have achievable probabilities and carefully selected functionality specifications. Thus, in such a system, the effects of damage will not be masked necessarily; and, provided the probabilities are met in practice, degraded or alternate service will occur. In effect, this implies that the survivability requirement will be achieved by the fault tolerance mechanism, i.e., the system will have a fault-tolerant design. Note, however, that the *N* different functions in the survivability specification do not correspond to functions that can be achieved with the resources that remain after *N* different faults. The *N* functions in the survivability specification are defined by application engineers to meet application needs and bear no prescribed relationship to the effects of faults. Many different faults might result in the same degraded or alternate application service.

In order to implement fault tolerance, the faults that have to be tolerated must be defined. A system cannot be built that merely tolerates “faults”. It is necessary to state what faults have to be tolerated because otherwise the damage that the fault causes cannot be predicted and hence treated. Precisely what faults are anticipated and what the system is required to do when specific faults occur, i.e., how the fault will be treated, must be defined so that the system builders know what states the system might enter and with what probability. This is crucial input to the process that has to lead to a design of a survivable system.

The analysis of faults has to be undertaken by application experts, hardware experts, security experts, disaster experts, and others. This information is essential application information and thus it is not within the purview of the computer system developers. Having to state what faults are likely to occur might seem counter-intuitive in the sense that damage is difficult to predict. But in fact the only way that appropriate responses can be defined is in the context of a known system state and that means a state in which the remaining resources are defined.

As a simple example, consider again a database system. Anticipated faults might include processor failure, disk failure, power failure, operator error, and perhaps flooding. In a multi-server environment it is possible to define a reasonable set of (reduced) application services that have to be maintained if a single server fails. If all the servers fail because of a flood, then it is likely that no service could be maintained. These are fundamentally different cases. Faults that were not defined as part of the system analysis, such as a gas explosion that destroys *unspecified* amounts of equipment, cannot possibly be followed with any degree of assurance by any function from the survivability specification. The system will not have been designed to cope with that particular fault.

7. Survivability and Security

Security attacks are a major concern for critical information systems, and in some discussions, survivability is viewed as synonymous with secure operation. Yet experience to date is that most significant service failures of critical information systems have been caused by things like erroneous software upgrades, operator mistakes, common-mode software faults, and not by security attacks. The damage that can result from a security attack can, of course, be tremendous but this is true with other types of fault also.

This is not to belittle the importance of security—clearly security attacks (i.e., deliberate faults) are a serious concern. What matters, however, is to address *all* anticipated types of fault including deliberate faults since it is the maintenance of customer value that we seek in survivability.

A survivable system is expected to continue to provide one of the forms of tolerable service after many different forms of damage have occurred. The anticipated faults for a given system frequently will include various types of malicious fault as well as all the other types, and so in developing the survivability specification and the associated system design, it is essential that a comprehensive approach be followed. A system that is secure yet is unavailable excessively because of hardware failures is not survivable.

Security is impacted by some aspects of design for dependability since the introduction of redundancy makes protection of a system from deliberate faults more difficult. For example, replication of data so as to provide a means of tolerating certain faults provides multiple opportunities for malicious data access. This presents formidable challenges in both specification and implementation of survivability.

8. Conclusions and Future Work

There are many critical information systems upon which many critical infrastructure applications rely. Loss of the information system will, in many cases, either reduce the service available from a critical infrastructure application or eliminate it entirely.

In this paper we have presented a formal definition of survivability, and related it to the field of dependability and the technology of fault tolerance. We claim that the specialized requirements of critical information systems require a new facet of dependability and that the survivability as we have defined it is different from reliability, availability, safety, and so on.

The definition that we have presented suggests other areas of investigation that we plan to pursue. In particular, the notion of value that we have used could be viewed differently as cost and the principle of providing value could be reformulated as minimizing cost where the preferred state would be the baseline. This approach has the benefit that classical optimization models of dynamic system behavior are often expressed in terms of cost minimization.

A second area of additional investigation is suggested by the dynamic nature of value and the need to develop more comprehensive analytic models of value as a function of time. This is an important component of a more comprehensive system model in which value is matched against demand. The ability to provide value (or, equivalently, minimize cost) is really a combination of service and its perceived utility. Thus demand enters the equation since it is demand that defines the utility of a service. A comprehensive model that accounts for both service and perceived value would allow much more precise tuning of the state changes that a system undertook when faults arose.

Acknowledgments

This effort was sponsored in part by the Defense Advanced Research Projects Agency and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-96-1-0314. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Rome Laboratory or the U.S. Government. This effort was sponsored in part by the National Science Foundation under grant number CCR-9804078.

References

[1] Anderson, T. and P. Lee. *Fault Tolerance: Principles and Practice*. Prentice Hall, Englewood Cliffs, NJ, 1981.

[2] Armstrong, J.H., *The Railroad: What It Is, What It Does*, Simmons-Boardman, Omaha, NE, 1993.

[3] Ball, R.E., *The Fundamentals of Aircraft Combat Survivability Analysis and Design*, American Institute of Aeronautics and Astronautics (AIAA), 1985.

[4] Deutsch, M.S. & Willis, R.R., *Software Quality Engineering: A Total Technical and Management Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1988.

[5] Ellison, B., D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead. “Survivable Network Systems: An Emerging Discipline,” Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University, November 1997.

[6] Knight, J., M. Elder, J. Flinn, and P. Marx. “Summaries of Four Critical Infrastructure Systems,” Technical Report CS-97-27, Department of Computer Science, University of Virginia, November 1997.

[7] Laprie, J. “Dependable Computing: Concepts, Limits, Challenges,” Special Issue FTCS-25: 25th International Symposium on Fault-Tolerant Computing, June 1995, pp. 42-54.

[8] Myers, J.F., “On Evaluating The Performability Of Degradable Computing Systems”, IEEE Trans. Computers, vol. C-29, no. 8, pp. 720-731, August 1980.

[9] Myers, J.F., W.H. Sanders, “Specification And Construction Of Performability Models” Proceedings: Second International Workshop on Performability Modeling of Computer and Communication Systems, Mont Saint-Michel, France, June 28-30, 1993.

[10] National Defense Industrial Association Symposium, Proceedings of Aircraft Survivability 2000, Monterey, CA, November 2000.

[11] Neumann, P.G., “Risks to the public in computer systems”, Software Engineering Notes, Vol 11, No 1, January 1986, pp 3-5.

[12] Office of the Undersecretary of Defense for Acquisition and Technology. *Report of the Defense Science*

Board Task Force on Information Warfare - Defense (IW-D), November 1996.

[13] President's Commission on Critical Infrastructure Protection. *Critical Foundations: Protecting America's Infrastructures The Report of the President's Commission on Critical Infrastructure Protection*, United States Government Printing Office (GPO), No. 040-000-00699-1, October 1997.

[14] Sullivan, K.J., J.C. Knight, X. Du, and S. Geist, "Information Survivability Control Systems", Proceedings of ICSE 21: Twenty First International Conference on Software Engineering, Los Angeles, CA, May 1999.

[15] Summers, B.J. (ed.), *The Payment System: Design Management and Supervision*, International Monetary Fund, Washington DC, 1994.

[16] U.S. Department of Commerce, National Telecommunications and Information Administration, Institute for Telecommunications Services, Federal Standard 1037C.

[17] http://www.cert.org/research/isw.html

[18] http://www.aircraft-survivability.com/