Availability, Reliability, and Survivability: An Introduction and Some Contractual Implications

Dr. Jack Murphy
DeXIsive Inc.

Dr. Thomas Ward Morgan
CACI Federal

This article is directed toward information technology professionals that enter into contractual agreements requiring service-level agreements (SLAs) that specify availability, reliability, or survivability objectives. Its purpose is to show a relationship between cost, performance, and SLA levels established by the customer.

Information technology (IT) outsourcing arrangements frequently employ service-level agreements (SLAs) that use terms such as availability and reliability. The intent is that the buyer requests a specific system availability and reliability (e.g., 98 percent to 99.9 percent, and 85 percent to 90 percent, respectively). The service provider is typically rewarded for exceeding specified limits and/or punished for falling below these limits.

In recent years, another term, survivability, has become popular and is used to express yet another objective: the ability of a system to continue functioning after the failure of one of its components. This article examines these terms so buyer and seller can understand and use them in a contractual context and designers/operators can choose optimal approaches to satisfying the SLAs.

The northeastern U.S. power grid failure in August 2003 drew attention to the availability, reliability, and survivability of business-critical IT systems. Catastrophe can be the catalyst for new thinking about the survivability of IT systems.

From the buyer's perspective, an increase in availability, reliability, and survivability comes at a price: 100 percent is not possible, but 98 percent might be affordable and adequate while 99.99 percent might be unaffordable or excessive. From the service provider's perspective, under-engineering or inadequate operating practices can result in penalties for failing to meet SLAs.

Availability

What Is Availability?

Availability is influenced by the following:

- **Component Reliability.** A measure of the expected time between component failures. Component reliability is affected by electromechanical failures as well as component-level software failure.
- **System Design.** The manner in which components are interrelated to satisfy required functionality and reliability. Designers can enhance availability through judicious use of redundancy in the arrangement of system components.
- **Operational Practices.** Operational practices come into play after the system is designed and implemented with selected components. Interestingly, after a system is designed, components are selected and the system is implemented. The only factor that can improve or degrade availability is operational practices.

Informally, system availability is the expected ratio of uptime to total elapsed time. More precisely, availability is the ratio of uptime and the sum of uptime, scheduled downtime, and unscheduled downtime:

$$A_1 = \frac{\text{Uptime}}{\text{Uptime + Downtime}} \qquad (1)$$

The formula (1) is useful for measuring availability over a given period of time such as a calendar quarter, but not very useful for predicting availability or engineering a system to satisfy availability requirements. For this purpose, system designers frequently employ a model based on mean time between failure (MTBF) and mean time to repair (MTTR), usually expressed in units of hours:

$$A_2 = \frac{\text{MTBF}}{\text{MTBF + MTTR}} \qquad (2)$$

Formula (2) is analogous to formula (1), but is based on statistical measures instead of direct observation. Most vendors publish MTBF data. MTTR data can often be collected from historical data. Interestingly, MTTR is partially within the control of the system operator. For example, the system operator may establish a strategy for spares or provide more training to the support staff to reduce the MTTR. Because of these factors, component vendors typically do not publish MTTRs. Finally, it should be noted that A_2 does not explicitly account for scheduled downtime.

The most common approach to include scheduled maintenance time is to include it in the total time represented in the mathematical model's denominator, thus reducing expected availability commensurately. This provides an additional challenge for the designer, but like MTTR it is somewhat controllable through operational procedures. If the system can be designed for only infrequent preventive maintenance, then availability is enhanced.

In most operational environments, a system is allowed to operate normally, including unscheduled outages, for some fixed period of time, *t*, after which it is brought down for maintenance for some small fraction of that time, λt (see Figure 1).

If the scheduled maintenance is periodic and on a predictable schedule, then following t hours there is a scheduled outage of λt so that the fraction of time that the system is not in maintenance is:

$$t / (t + \lambda t) = 1 / (1 + \lambda) \qquad (3)$$

Since the denominator in A2 does not include the time in preventive maintenance, an adjustment to formula (1) is needed whenever preventive maintenance is part of the operational routine. To include this time, the denominator needs to be increased by a factor of 1+ λ to accurately reflect the smaller actual availability expected. Stated differently, the denominator in A_2 needs to be modified to accurately represent all time, including operational (MTBF), in repair (MTTR), or in maintenance λ (MTBF + MTTR). The revised availability model in cases of scheduled downtime is:

$$A_3 = \frac{\text{MTBF}}{(1 + \lambda)(\text{MTBF + MTTR})} \qquad (4)$$

This model, like the model A_2, assumes independence between the model variables. In reality there may be some relationship between the variables MTBF, MTTR, and λ.

Availability Engineering

A complex system is composed of many interrelated components; failure of one component may not impact availability if the system is designed to withstand such a failure, while failure of another component may cause system downtime and hence degradation in availability. Consequently, system design can be used to achieve high availability despite unreliable components.

For example, if Entity1 has component availability 0.9 and Entity2 has component availability 0.6, then the system availability depends on how the entities are arranged in the system. As shown in Figure 2, the availability of System1 is dependent on the availability of both Entity1 and Entity2.

In Figure 3, the system is unavailable only when both components fail. To compute the overall availability of complex systems involving both serially dependent and parallel components, a simple recursive use of the formulas in Figures 2 and 3 can be employed.

Thus far, the engineer has two tools to achieve availability requirements specified in SLAs: a selection of reliable components and system design techniques. With these two tools, the system designer can achieve almost any desired availability, albeit at added cost and complexity.

The system designer must consider a third component of availability: operational practices. Reliability benchmarks have shown that 70 percent of reliability issues are operations induced versus the traditional electromechanical failures. In another study, more than 50 percent of Internet site outage events were a direct result of operational failures, and almost 60 percent of public-switched telephone network outages were caused by such failures [1]. Finally, Cisco asserts that 80 percent of non-availability occurs because of failures in operational practices [2]. One thing is clear: Only a change in operational practices can improve availability after the system components have been purchased and the system is implemented as designed.

In many cases, operational practices are within control of the service provider while product choice and system design are outside of its control. Conversely, a system designer often has little or no control over the operational practices. Consequently, if a project specifies a certain availability requirement, say 99.9 percent (3-nines), the system architect must design the system with more than 3-nines of availability to leave enough *head space* for operational errors.

To develop a model for overall availability, it is useful to consider failure rate instead of MTBF. Let α denote the failure rate due to component failure only. Then $\alpha = 1/\text{MTBF}$. Also, let τ denote the total failure rate, including component failure as well as failure due to operational errors. Then $1/\tau$ is the mean time between failure when both component failure and failures due to operational errors are considered (MTBF_{Tot}).

If β denotes the fraction of outages that are operations related, then $(1 - \beta)\ \tau$ is the fraction of outages that are due to component failure. Thus:

$$\begin{gathered}(1 - \beta)\ \tau = \alpha \\ \text{So } \tau = \alpha / (1 - \beta) \text{ and} \\ \text{MTBF}_{\text{Tot}} = (1 - \beta) / \alpha \end{gathered} \tag{5}$$

The revised model becomes:

$$A_4 = \frac{\text{MTBF}_{\text{Tot}}}{(1 + \lambda)(\text{MTBF}_{\text{Tot}} + \text{MTTR})} \tag{6}$$

where,

$$\text{MTBF}_{\text{Tot}} = \frac{1 - \beta}{\alpha}$$

When an SLA specifies an availability of 99.9 percent, the buyer typically assumes the service provider considers all forms of outage, including component failure, scheduled maintenance outage, and outages due to operational error. So the buyer has in mind a model like that defined by A_4. But the designer typically has in mind a model like A_2 because the design engineer seldom has control over the maintenance outages or operationally induced outages, but does have control over product selection and system design. Thus, the buyer is frequently disappointed by insufficient availability and the service provider is frustrated because the SLAs are difficult or impossible to achieve at the contract price.

If the system design engineer is given insight into λ, the maintenance overhead factor, and β, then A_2 can be accurately determined so that A_4 is within the SLA. For example, if A_4 = 99 percent, it may be necessary for the design engineer to build a system with A_2 = 99.999 percent availability to leave sufficient room for maintenance outages and outages due to operational errors.

Given an overall availability requirement (A_4) and information about λ and β, the design availability A_2 can be computed from formula (7). Note that high maintenance ratios become a limiting factor in being able to engineer adequate availability.

$$A_2 = \frac{(1 + \lambda)A_4}{1 + \beta\ ((1 + \lambda)\ A_4 - 1)} \tag{7}$$

For 3-nines of overall availability it is necessary to engineer a system for over 6-nines of availability (less than 20 seconds/year downtime due to component failure) even if only 50 percent of outages are the result of operational errors when the maintenance overhead is 0.1 percent. Engineering systems for 6-nines of availability may have a dramatic impact on system cost and complexity. It may be better to develop operational practices that minimize repair time and scheduled maintenance time.

Reliability

What Is Reliability?

There is an important distinction between the notion of availability presented in the preceding section and reliability. Availability is the expected fraction of time that a system is operational. Reliability is the probability that a system will be available (i.e., will not fail) over some period of time, *t*. It does not measure or model downtime. Instead reliability only models the time until failure occurs without concern for the time to repair or return to service.

Reliability Engineering

To model reliability, it is necessary to know something about the failure stochastic process, that is, the probability of failure before time, *t*. The Poisson Process, based upon the exponential probability distribution, is usually a good model. For

Figure 1: *Preventative Maintenance Cycle*

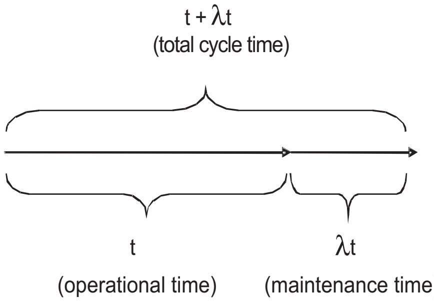


Figure 2: *Availability With Serial Components*

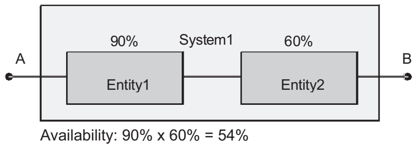
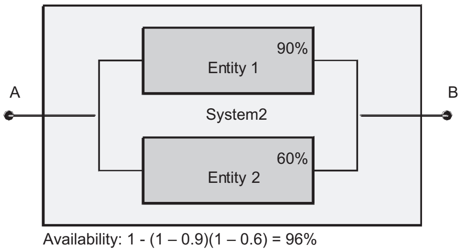


Figure 3: *Availability With Parallel Components*



this process, it is only necessary to estimate the mean of the exponential distribution to predict reliability over any given time interval. Figure 4 depicts the exponential distribution function and the related density function. As shown, the probability of a failure approaches 1 as the period of time increases.

If F(t) and f(t) are the exponential distribution and density functions respectively, then the reliability function R(t) = 1 – F(t). So that,

$$\mathbf{R(t) = 1- \int_t^\infty \mathit{f}(t) = \int_0^t \mathit{f}(t) = \int_0^t \frac{1}{\theta} = e^{-t/\theta}} \qquad \mathbf{(8)}$$

where,

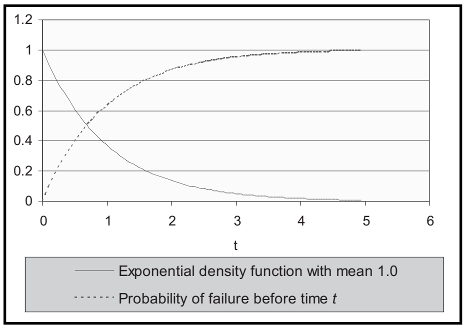
$$\theta = \mathbf{MTBF}$$

The mean of this distribution is θ, the MTBF. It can be measured directly from empirical observations over some historical period of observation or estimated using the availability models presented earlier.

Table 1 shows the reliability for various values of θ and *t*. Note that when t = MTBF, R(t) = 36.79 percent. That is, whatever the MTBF, the reliability over that same time period is always 36.79 percent.

Network components such as Ethernet switches typically have an MTBF of approximately 50,000 hours (about 70 months). Thus the annual reliability of a single component is about 85 percent (use t=12 and θ=70 in the formula above or interpolate using Table 2). If that component is a single point of failure from the perspective of an end workstation, then based on component failure alone the probability of outage for such workstations is at least 15 percent. When operational errors are considered, it is $MTBF_{Tot}$, not MTBF that determines reliability, so the probability of an unplanned outage within a year's time increases accordingly.

Figure 4: *Exponential Density and Distribution Functions With MTBF = 1.0*



The preceding assumes that the exponential density function accurately models system behavior. For systems with periodic scheduled downtime, this assumption is invalid. At a discrete point in time, there is a certainty that such a system will be unavailable: $R(t) = e^{-t/\theta}$ for any $t < t_0$, where t_0 is the point in time of the next scheduled maintenance, and $R(t) = 0$ for $t \geq t_0$.

Survivability

What Is Survivability?

Survivability of IT systems is a significant concern, particularly among critical infrastructure providers. Availability and reliability analysis assume that failures are somewhat random and the engineer's job is to design a system that is robust in the face of random failure. There is thus an implicit assumption that system failure is largely preventable.

Survivability analysis implicitly makes the conservative assumption that failure will occur and that the outcome of the failure could negatively impact a large segment of the subscribers to the IT infrastructure. Such failures could be the result of deliberate, malicious attacks against the infrastructure by an adversary, or they could be the result of natural phenomenon such as catastrophic weather events. Regardless of the cause, survivability analysis assumes that such events can and will occur and the impact to the IT infrastructure and those who depend on it will be significant.

Survivability has been defined as "the capability of a system to fulfill its mission in a timely manner, in the presence of attacks, failures, or accidents" [3]. Survivability analysis is influenced by several important principles:

- **Containment.** Systems should be designed to minimize mission impact by containing the failure geographically or logically.
- **Reconstitution.** System designers should consider the time, effort, and skills required to restore essential mission-critical IT infrastructure after a catastrophic event.
- **Diversity.** Systems that are based on multiple technologies, vendors, locations, or modes of operation could provide a degree of immunity to attacks, especially those targeted at only one aspect of the system.
- **Continuity.** It is the business of mission-critical functions that they must continue in the event of a catastrophic event, not any specific aspect of the IT infrastructure.

Table 1: *Reliability Over t Months For MTBF Ranging From 3 to 96 Months*

R(t)		t (months)							
		1	2	4	8	12	24	48	96
MTBF (months)	3	71.65%	51.34%	26.36%	6.95%	1.83%	0.03%	0.00%	0.00%
	6	84.65%	71.65%	51.34%	26.36%	13.53%	1.83%	0.03%	0.00%
	12	92.00%	84.65%	71.65%	51.34%	36.79%	13.53%	1.83%	0.03%
	24	95.92%	92.00%	84.65%	71.65%	60.65%	36.79%	13.53%	1.83%
	48	97.94%	95.92%	92.00%	84.65%	77.88%	60.65%	36.79%	13.53%
	96	98.96%	97.94%	95.92%	92.00%	88.25%	77.88%	60.65%	36.79%

If critical functions are composed of both IT infrastructure (network) and function-specific technology components (servers), then both must be designed to be survivable. An enterprise IT infrastructure can be designed to be survivable, but unless the function-specific technologies are also survivable, irrecoverable failure could result.

Measuring Survivability

From the designers' and the buyers' perspectives, comparing various designs based upon their survivability is critical for making cost and benefit tradeoffs. Next we discuss several types of analysis that can be performed on a network design that can provide a more quantitative assessment of survivability.

Residual measures for an IT infrastructure are the same measures used to describe the infrastructure before a catastrophic event but are applied to the expected state of the infrastructure after the effects of the event are taken into consideration. Here we discuss four residual measures that are usually important:

- **Residual Single Points of Failure.** In comparing two candidate infrastructure designs, the design with fewer single points of failure is generally considered more robust than the alternative. When examining the survivability of an infrastructure with respect to a particular catastrophic event, the infrastructure with the fewer residual single points of failure is intuitively more survivable. This measure is a simple count.
- **Residual Availability.** The same availability analysis done on an undamaged infrastructure can be applied to an infrastructure after it has been damaged by a catastrophic event. Generally, the higher the residual availability of an infrastructure the more survivable it is with respect to the event being analyzed.
- **Residual Performance.** A residual infrastructure that has no single point of failure and has high residual availability may not be usable from the per-

spective of the surviving subscribers (users). Consequently, the performance received by the surviving user community needs to be analyzed. The analysis must take into consideration any increase or decrease in infrastructure activity resulting from the organizational response to the event being studied. As an example, the performance of file transfers across an enterprise may average 100 megabits per second (Mbps) under normal circumstances but a catastrophic failure may reduce the performance to 10 Mbps even if there is no loss of service (i.e., availability).

- **Reconstitution Time.** Once a catastrophic event has taken place, the time required to resume mission-critical activities is one of the most important residual measures for describing an IT infrastructure's survivability. Calculations of reconstitution time should take into consideration both emergency recovery plans and impairment of recovery capabilities caused by the event.

Comparing Architectures

In evaluating alternative architectures, some composite measures across a set of potential events are often useful. In this section, we suggest two methods to compare the survivability of alternative architectures.

Develop a Conical Event Set

A conical set of events is a set that includes all the important types of catastrophic events that the IT infrastructure should be designed to survive. For example, if fires, floods, viruses, and power outages are the types of events that must be anticipated in the IT infrastructure design, then the conical set of events includes at least one example of each type. Ideally, the conical set of events includes the worst case example of each event type.

Compare the Survivability of Architectures

After calculating the residual metrics for each of the events in the conical set and alternative architectures, it may be desirable to make an objective comparison of the survivabilities. We discuss two conceptual ways of making such comparisons.

- **Multiple Criteria Methods.** Within the specialty area of multi-criteria optimization, the concept of best alternatives has been formalized. This concept simply states that if one alternative has scores that are greater or equal to the corresponding scores of all other alternatives and has a unique best score for at least one criterion, it is clearly the best alternative. Consequently, if one of the architectures being evaluated has higher residual availability for all events, lower reconstitution time for all events, and fewer single points of failures for all events, it is clearly the best architecture among those being compared.
- **Weighting Methods.** In situations where there are many criteria that can be weighted by some subjective means and no clear best alternative based upon multiple criteria methods exists, a reasonable approach to selecting the most survivable alternative is to create a survivability index. The index would be a simple, weighted sum of the criteria c_{ij} values for each catastrophic event multiplied by their respective weights, w_{ij}. An index value is computed for each alternative and the alternative with the highest (or lowest) index is selected as the best alternative. The formula is:

$$\text{Survivability Index} = \sum_i \sum_j w_{i,j}\, c_{i,j} \qquad (9)$$

In (9), the *i* index ranges over the set of catastrophic events, and the *j* index ranges over the survivability criteria.

Summary

Availability and reliability are well-established disciplines upon which SLAs are frequently established. However, survivability is an increasingly important factor in the design of complex systems. More effort is needed for survivability to achieve the same rigor as enjoyed by availability and reliability.◆

References

1. Patterson, D., et al. "Recovery Oriented Computing: Motivation, Definition, Techniques, and Case Studies." Berkeley, CA: University of California Berkeley, Mar. 2002.
2. Cisco Systems. "Service Level Management Best Practices White Paper." San Jose, CA: Cisco Systems, July 2003.
3. Ellison, R.J., et al. "Survivable Network Systems, an Emerging Discipline." Technical Report CMU/SEI-97-TR-013, 1997.

About the Authors

Jack Murphy, Ph.D., is president and chief executive officer of DeXIsive Inc., an information technology system integrator focusing on information assurance, network solutions, and enterprise infrastructure applications. Prior to this, Murphy was the chief technical officer for EDS U.S. Government Solutions Group. He is an EDS Fellow Emeritus and remains actively involved with EDS thought leaders. Murphy retired as a lieutenant colonel from the Air Force where he spent much of his career on the faculty of the Air Force Academy teaching math, operations research, and computer science. He has a doctorate in computer science from the University of Maryland.

DeXIsive, Inc.
4031 University DR STE 200
Fairfax, VA 22030
Phone: (703) 934-2030
Cell: (703) 867-1246
E-mail: jack.murphy@dexisive.com

Thomas Ward Morgan, Ph.D., is a chief simulation scientist at CACI International, and leads the World Wide Engineering Support Group Modeling and Simulation team. He has been involved in networking and software development since 1970. Morgan served in the Army Medical Service Corps and worked on the Automated Military Outpatient System and Next Generation Military Hospital projects. He has also worked at AT&T Bell Laboratories participating in the design of internal corporate networks to support AT&T's customer premises equipment operations. He was active in the Institute for Operations Research and the Management Sciences and has published over 60 technical papers. He has a Master of Science and a doctorate in electrical engineering.

CACI Federal
14111 Park Meadow DR STE 200
Chantilly, VA 20151
Phone: (703) 802-8528
E-mail: wmorgan@caci.com