Mission-Based System Reliability Modeling for Establishing Testable Performance Requirements of a Distributed Network Monitoring System

Arthur Fries[1] and Garfield Jones[2]

•

[1]Institute for Defense Analyses, Alexandria, Virginia 22311, USA, afries@ida.org
[2]Morgan State University, Baltimore, Maryland, 21251, USA, Garfield.jones@morgan.edu

Abstract

Mission-based subsystem reliability requirements are derived for a parent distributed network monitoring system operating under circumstances that differ from standard analytical constructs in a number of ways. First, the system comprises a hierarchy of elements of different functionalities individually adhering to distinct operational profiles. Second, some constituent elements only need to perform during relatively small and non-predetermined portions of the overall system mission accomplishment window. Third, failed elements can be restored or replaced in time to enable additional opportunities for satisfying mission needs.

Keywords: Distributed Network Monitoring System, Subsystem reliability, Operational profile, Mean time between operational mission failures

I. Introduction

A distributed network monitoring system (DNMS) is to be integrated into the current architecture of an existing computer network supporting operations across an extensively dispersed organization. The DNMS will provide the capability to regularly check and report on the security posture of the devices on the parent network. A challenge is to establish credible performance requirements for the constituent elements of the DNMS – to aid design and implementation planning, and to enable reliability demonstration analyses that can accommodate historical DNMS element reliability data as well as dedicated DNMS test results at both the element and system levels. To that end, this paper formulates tractable analytical models that plausibly represent anticipated DNMS operational and maintenance profiles, which vary by DNMS element type, and link DNMS mission performance specifications, as prescribed by organization management, to reliability requirements for the individual classes of DNMS elements.

This setting deviates from standard calculations of system reliability requirements in three fundamental ways. First, the DNMS comprises a hierarchy of subordinate elements of different functionalities individually adhering to distinct mission-based operational profiles. Second, some constituent elements only need to perform during relatively small and non-predetermined portions of the overall DNMS mission accomplishment window. Third, a failed element can be restored or replaced in time to enable additional chances for satisfying DNMS mission needs, with the number of opportunities depending on sub-system and prevailing network support processes.

While much of the operational functionality of a DNMS element is software centric, the composition of DNMS elements includes both dedicated hardware and software whose configurations, both in numbers and design, contribute to the reliability ascribed to particular ensembles. For example, higher quality parts and/or redundancy of supporting integral equipment and operational processes can be built in to enhance system reliability. Accordingly, it is appropriate to pursue more traditional reliability formulations [1, 2] vice focusing on software engineering perspectives [3].

Section II sketches the general structure of a DNMS. Section III describes associated operational and maintenance profiles and translates them into tractable reliability modeling approaches. The discussions presented in Section IV elaborate on the analytical constructs and outline potential follow-on and related reliability analyses.

II. Architecture

The notional DNMS depicted in Figure 1 is composed conceptually of four constituent element types:

1. Individual automated sensors that scan network hardware and software objects for specified defects. Different types of sensors search for distinctive classes of network defects. For each type of sensor, multiple copies are needed to scan the entire network in a reasonably time-efficient manner.
2. A data interface and integration layer that standardizes, processes, and transmits information collected by the automated sensors to base-level dashboards.
3. Base dashboards that process local network scanning data and display aggregated statistics to attendant network security monitors and administrators.
4. A master dashboard that encapsulates summaries from lower level dashboards and enables top-level organization management to track the security posture across the entire network. A back-up master dashboard, operating in a warm standby mode, provides redundancy.

Note that the execution steps essential to DNMS performance are mutually independent across the four layers. Further, within any given layer there are no dependencies among individual elements.

The domain for a single base dashboard encompasses a natural subdivision of the network, e.g., a particular division, component, agency, sub-organization, or geographical location. In addition to receiving data from subordinate dashboards, the master dashboard supports communications down to lower level dashboards and the associated staff. The redundancy provided by the back-up master dashboard enhances organization leadership's access to DNMS information at any critical time point. Dashboards cannot continuously provide real-time status reports for the whole organization, as that would necessitate constant sensor scanning across the entire network. Some acceptable data latency period (e.g., less than a nominal number of prescribed business days) is tolerable and is reflected in DNMS operating profiles.

From the user viewpoint, the new DNMS, while adding modestly to the day-to-day operational mission workload of the parent organization, enhances existing network security processes. In particular, dashboard displays illuminate categories of detected network security defects and characterize their incidence and distribution across the network. These promote the development of mitigation strategies and prioritized implementations, both at the overall and localized network levels.

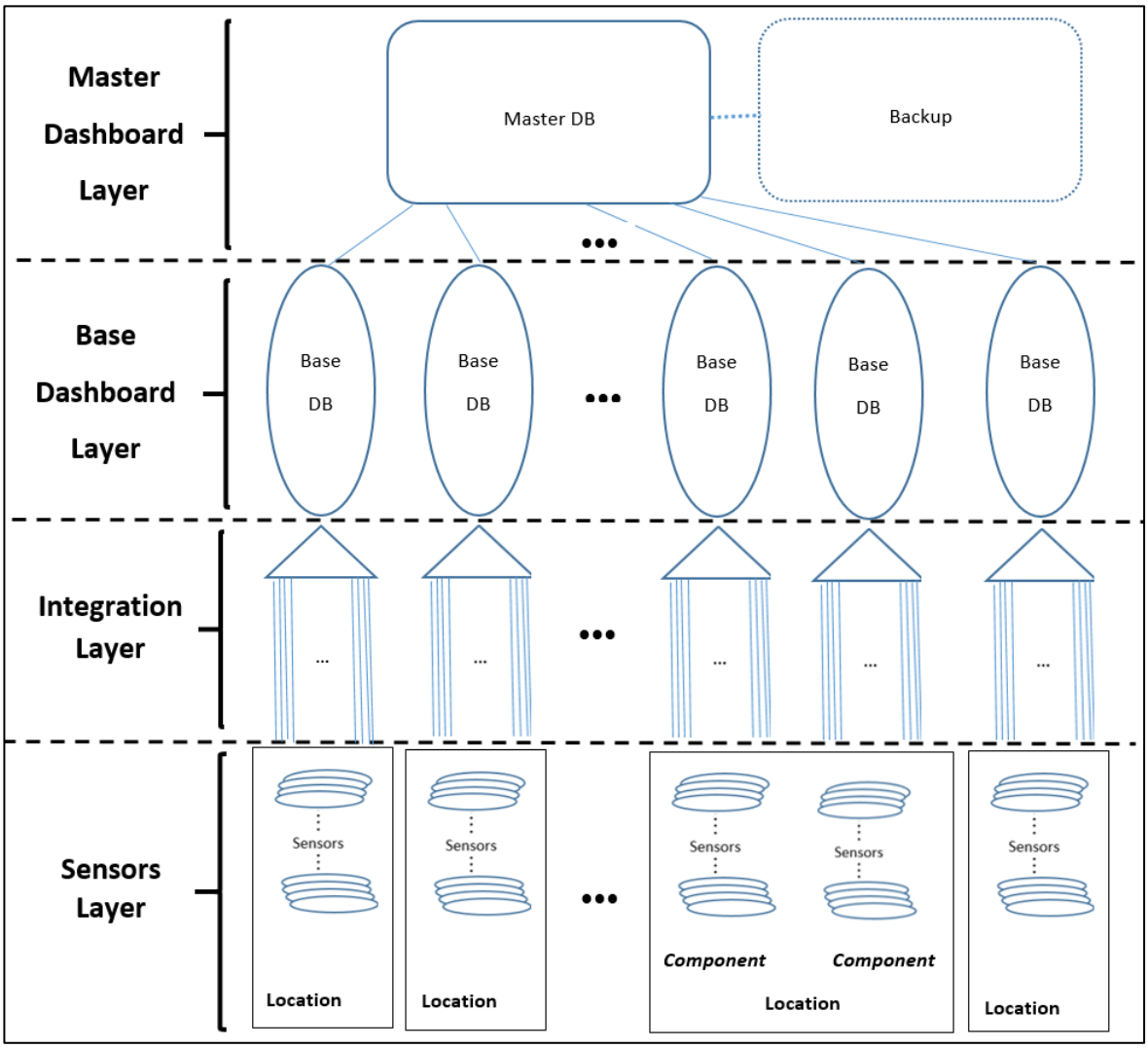

Figure 1: *Notional DNMS structure*

III. Modeling

The modeling approaches presented here are simplistic, favoring analytical tractability and ease of exposition. (Section IV offers additional discussions.) For any real world application, one could of course incorporate the specifics that characterize the subject network.

The operational mission of a DNMS is to systematically monitor and regularly report on the cybersecurity health of the organization's network. This entails regularly scanning the network, updating detailed data on detected defects, and summarizing results in dashboard displays. All of these activities are to be accomplished every d days within the backdrop of ongoing organization business activities. The organization-level requirement is that with some prescribed high probability, P, the DNMS will successfully complete each of its fundamental mission functions within any d-day operational window. For large networks, nominal values of d might be as long as 5-7 days or as short as 2-3 days. To support DNMS design decompositions and prospects for reliability inference based on disparate information sources, the mission success probability P is parsed into subsystem and component element performance requirements. Begin by writing $P = P_1 P_2 P_3 P_4$, where each distinct P_i, $i = 1,2,3,4$, is the probability that the i-th level of the DNMS (as defined in Section II and portrayed

In Figure 1) will successfully execute each of its mission essential functions within an operational performance window. Imposing equal apportionment translates to setting $P_1 = P_2 = P_3 = P_4$ and obtaining $P_i = P^{1/4}$, $i = 1,2,3,4$. This simplification enjoys the practical advantage of framing subsequent calculations in terms of a single parameter to be provided by the organization's management. The subsequent derivations examine each DNMS level separately, consider relevant operational and maintenance profiles, and obtain associated performance probability and element reliability requirements. These are translated to specifications of mean time between operational mission failures (MTBOMF) for individual DNMS element types, i.e., reliability requirements, a format that is more amenable for classes of subsystem design and testing analyses.

III.i. Sensors

A single sensor is tasked to scan a designated portion of the entire network sometime within each d-day performance window, record the collected data, and disseminate to the integration layer. These primary functions must be accomplished in time to allowing adequate opportunity for the integration layer and dashboards to complete their related data processing within that same d-day span. Accordingly, it is plausible to assume that the sensor undertakes an initial execution attempt relatively early within the operational performance window, the probability of it being successful in its initial foray is p_1, and that success entails operating without fault for t_1 consecutive hours (small compared to $24d$ hours). If the initial attempt is successful, no further sensor operation is required until the next d-day performance window arrives. If unsuccessful, a failure event, or lack of a success event, is automatically registered and diagnostic steps are initiated to determine the failure cause and restore or replace the sensor to an as good as new state. This conceptually includes the possibility of temporarily reassigning another sensor to complete the subject sensor's original obligations. The subsequent attempt likewise may be successful or fail, with the same probabilistic characteristics. It is assumed that the value of t_1 and the capabilities of the DNMS-specific logistical support processes, including consideration of availability and restoration times, could enable up to a_1 attempts for the sensor to complete a suitable execution within the desired timeframe. Nominal values of a_1 may be in the vicinity of $d/2$.

The probability that the sensor successfully completes its operational mission takes the form

$$1 - (1 - p_1)^{a_1} = 1 - \left(1 - e^{-t_1/\theta_1}\right)^{a_1}, \quad (1)$$

upon imposing a standard exponential time to failure distribution and set the associated MTBOMF value equal to θ_1. This result holds for a single sensor. The DNMS, however, comprises different sensor types and varying counts for each. Say the total number of sensors is n_1 (which could be hundreds for a large network). Treating their behaviors as being identical and independent, the probability that the entire sensor layer successfully executes a d-day operational mission is

$$\left[1 - \left(1 - e^{-t_1/\theta_1}\right)^{a_1}\right]^{n_1}. \quad (2)$$

Equating this to the prescribed DNMS mission performance requirement of $P_1 = P^{1/4}$, one deduces the associated MTBOMF value, i.e., the reliability requirement for a DNMS sensor element:

$$\theta_1 = -t_1 / ln\left[1 - \left(1 - P^{1/4n_1}\right)^{1/a_1}\right]. \quad (3)$$

(Some adhere to the common definition that reliability is the probability that an item will perform its intended function for a specified time interval under stated conditions [4] – which, in this paper's setting, aligns more closely to the formulation $p_1 = e^{-t_1/\theta_1}$.) For the representative set of values $t_1 = 4$ hours, $P = 0.999$, $n_1 = 500$, and $a_1 = 2$, (3) yields $\theta_1 = 5654$ hours.

III.ii. Integration Layer

Three different concepts for how a DNMS integration layer may be structured are considered here. Each description is accompanied by its own derivation of the associated MTBOMF requirement ascribed to a single resident element.

One conceivable structure of an integration layer, Design I, connects each sensor via a directed pathway to its assigned base dashboard. There are $n_2 = n_1$ such conduits, each determining an individual integration layer element, and the parameter definitions and logic underlying the development of (3) transfer straightforwardly to obtain

$$\theta_{2,\mathrm{I}} = -t_2/ln\left[1-\left(1-P^{(1/4n_2)}\right)^{1/a_2}\right]. \quad (4)$$

Considering the values $t_2 = 4$ hours, $P = 0.999$, $n_2 = 500$, and $a_2 = 3$, which vary from their sensor level counterparts only in that the number of attempts has been increased from 2 to 3, (4) leads to a reduced MTBOMF requirement of $\theta_{2,\mathrm{I}} = 502$ hours.

A variation of the preceding construct incorporates a set of additional elements, data aggregation devices, one interfacing with each unique base dashboard. The data flow corresponding to Design II is sensor ⇨ pathway ⇨ data aggregation device ⇨ base dashboard. The equal apportionment principle allocates a mission success probability of $P_2^{1/2}$ to each class of elements in Design II. For an individual conduit element, the required MTBOMF threshold thus can be read directly from (4):

$$\theta_{2,\mathrm{II(c)}} = -t_2/ln\left[1-\left(1-P^{(1/8n_2)}\right)^{1/a_2}\right]. \quad (5)$$

Retaining the input specifications from the immediately preceding numerical example, insertion into (5) leads to the higher MTBOMF requirement of $\theta_{2,\mathrm{II(c)}} =$ 633 hours (consistent with the notion that the P in (4) effectively is increased to $P^{1/2}$ in (5)). For the data aggregation devices, the appropriate count of elements is n_3, the number of base dashboards (which is substantially smaller than n_2). Thus the associated MTBOMF requirement for a single data aggregator is simply

$$\theta_{2,\mathrm{II(a)}} = -t_2/ln\left[1-\left(1-P^{(1/8n_3)}\right)^{1/a_2}\right]. \quad (6)$$

For illustration purposes consider the same set of input parameters as in the two preceding examples, but substantially reduce the number of elements down by two orders of magnitude to $n_3 = 5$. The resultant MTBOMF requirement declines considerably to $\theta_{2,\mathrm{II(a)}} = 135$ hours. The pairing (5) and (6) assume that their values for the functional operational times and numbers of attempts available within the operational performance window are identical to their respective counterparts in (4). If need be, these can be adjusted appropriately.

Design III retains the presence of the data aggregation devices and accompanying assumptions, but excludes the antecedent pathways. Accordingly, the form of (4) holds and revising the relevant count of elements yields

$$\theta_{2,\mathrm{III}} = -t_2/ln\left[1-\left(1-P^{(1/4n_3)}\right)^{1/a_2}\right]. \quad (7)$$

Relative to (6), the power of P has been increased by a factor of two and the value of $\theta_{2,\mathrm{III}}$ will decrease commensurably. For the identical parameterization, the application of (7) gives $\theta_{2,\mathrm{III}} =$ 107 hours.

III.iii. Base Dashboards

For modeling purposes, the layout of base dashboards parallels that of Design III for the integration layer – as there is a one-to-one correspondence between data aggregation devices and base dashboards. Rewriting (7) to allow for possible changes in operations times and allowable number of tries to complete those operations, it follows that

$$\theta_3 = -t_3/ln\left[1-\left(1-P^{(1/4n_3)}\right)^{1/a_3}\right], \quad (8)$$

which differs from (7) merely by the multiplicative factor t_3/t_2. Here the value of t_3 includes the time needed to ingest the data from the integration layer as well as system on-time for displaying data summaries and supporting user needs. Setting $t_3 = 10$ hours (2.5 times t_2), $P = 0.999$, $n_3 = 5$, and $a_3 = 3$, (8) yields a MTBOMF requirement of $\theta_3 = 267$ hours – an increase of 150 percent compared to the comparable value given for (7).

III.iv. Master Dashboard

The operational profile for the master dashboard includes ingesting summary level data from each of the base dashboards, updating the backup master dashboard with that content, enabling bilateral information flows with the subordinate dashboards, and supporting continuous monitoring of the state of cybersecurity across the entire organization. The associated number of operating hours is t_4 hours per business day, totaling $t_4 d$ hours over a d-day performance window. A nominal value for t_4 is 10 hours. If the master dashboard loses some essential functionality, the backup master dashboard will be fully activated to serve as a substitute and maintain operations. Since the backup is running in a warm standby mode, the timing and nature of the manifested failure will determine whether the up-to-date summary data already has been mirrored in the backup, can be transferred from the "failed" master dashboard to the backup, or needs to be ingested anew by the backup.

A pragmatic perspective, consistent with the explicit design choice of a warm standby backup vice a hot standby, would not consider a one to two hour period for users of the master dashboard being deprived of wholly updated summary data as constituting an operational mission failure (OMF). The corresponding likelihood of mission success is the Poisson probability of no more than one failure occurring over the prescribed mission time, and the associated MTBOMF requirement value is the unique solution to the equation

$$P^{1/4} = e^{-(t_4 d/\theta_4)}[1 + (t_4 d/\theta_4)]. \quad (9)$$

The right-hand-side of (9) is the standard formula for hot standby reliability [5], and is appropriate here under the relaxed interpretation of a master dashboard OMF. For the values $P = 0.999$, $t_4 =$ 10 hours, and $d = 5$ days, (9) yields $\theta_1 = 2219$ hours.

To support the development of a model representation that accommodates broader definitions of OMFs, the parameter α is introduced to denote the probability that when operationalized the backup master dashboard need not ingest updated summary from the base dashboards. Additionally, a harsher definition of master dashboard success is imposed, demanding no break in the currency of data summary presentations. Under this construct, a simple generalization of (9) follows:

$$P^{1/4} = e^{-(t_4 d/\theta_4)}[1 + \alpha(t_4 d/\theta_4)]. \quad (10)$$

The limiting value $\alpha = 1$ recovers (9), while the other extreme $\alpha = 0$ gives no credit whatsoever for redundancy. Setting $\alpha = 0.5$ and retaining the immediately preceding example inputs, (10) determines a substantially higher MTBOMF requirement of $\theta_1 = 99{,}963$ hours. Even for $\alpha = 0.9$, the calculated MTBOMF is 20191 hours, more than nine times the corresponding threshold

presented earlier for (9). Clearly the stricter interpretation of a master dashboard OMF establishes considerably

higher reliability requirements and could motivate transitioning to a hot standby design.

III.v. Combined Dashboards Perspective

Under some circumstances, it may be reasonable in reliability calculations to treat the base and master dashboards as being identical. Relevant considerations include commonality of software platforms, software modules, and hardware components, and similarity of failure mode histories. When plausible, paired equations from III.3 and III.4 can be consolidated into a single equation representative of dashboards as a whole – after reapportionment of the DNMS mission success probability. For example, combining (9) with the appropriate transformation of (8) leads to the formulation

$$P^{1/2} = \left[1 - \left(1 - e^{-t_3/\theta_{3,4}}\right)^{a_3}\right]^{n_3} e^{-(t_4 d/\theta_{3,4})}\left[1 + \left(t_4 d/\theta_{3,4}\right)\right], \quad (11)$$

where the new notation $\theta_{3,4}$ denotes the common MTBOMF value ascribed to all of the dashboards. As the right-hand-side is a monotonically increasing function of $\theta_{3,4}$, (11) possesses a unique solution. From the collection of example input values presented earlier, $P = 0.999$, $t_3 = 10$, $a_3 = 3$, $n_3 = 5$, $t_4 = 10$, and $d = 5$, it follows that $\theta_{3,4} = 1567$ hours. This determination of the MTBOMF requirement lies between the two separate MTBOMF requirements calculated previously for (8) and (9), but is considerably closer to the latter, i.e., the influence of the master dashboard dominates. This would be even more so the case if the role of (9) in this example were to be replaced by the more demanding (10).

IV. Discussion

This paper develops tractable models for the reliability of a DNMS architecture comprised of four distinct levels with varying operational and maintenance profiles. They offer informative insights to contractors responsible for proposing, designing, deploying, and supporting the DNMS, seeking to balance design and operational implementation investments against formally prescribed system performance demands or possibly even subject to potential monetary penalties were the deployed DNMS to incur operational performance shortfalls. The straightforward model representations also can be utilized by DNMS host organizations to establish formal reliability demonstration requirements and to guide the development of operational and logistical support processes.

Additionally, both integrators and customers can utilize the framework to assess emerging reliability data from a deployed DNMS and to contemplate specific types of potential design refinements, both architectural and procedural. When conducting dedicated reliability demonstrations or scoring emerging results, care must be taken in defining what constitutes an OMF. For example, minor technical glitches that are nearly immediately remedied via automated or manually induced system reboots may be practically inconsequential. Also, follow-on DNMS integration activities naturally will occur over the deployed lifetime of the DNMS (e.g., coincident with changes to the organization's landscape or the application of routine software upgrades for individual classes of DNMS elements), and these can be expected to engender some initial sets of misconfiguration problems and start-up failures. Whether these should count as OMFs against the DNMS or as a separate category of system failures may depend on the purpose of the immediate reliability analyses and the specifics of any relevant formal requirements contractually imposed on DNMS integrator teams.

An alternative to relying on simple models of the type that constructed in this paper would be to pursue detailed simulation modeling of end-to-end DNMS performance steps over the

course of a subject d-day performance window. In addition to incurring requisite time and resource costs, such an approach would be confronted by several analytical challenges. First, definitive operating

profiles cannot be readily discerned. Recall that DNMS is an addition to an organization's existing functionalities and primary operational missions. From that perspective, the daily implementation of DNMS is of secondary importance and there are numerous options, depending on the organization's current operational priorities, of when and how DNMS will be activated and utilized during a particular d-day cycle. Likewise, maintenance events integral to DNMS diagnostic, replace, and restore processes cannot be precisely characterized.

The modeling constructs in this paper account for DNMS employment uncertainties via simplified but plausible representations that embrace DNMS implementation realities. The emphasis is on total operational time for each DNMS element type, vice detailed event-to-event sequencing. Further, the derivations focus on the number of attempts available to an element for completing its assigned operational mission, instead of modeling the detailed specifics of how logistical support processes enable multiple tries to be realized. This is compatible with conventional expressions of operational maintenance requirements (e.g., resolve help desk tickets by the end of the next business day) and can embrace formulations of a spectrum of support responsiveness. To pursue analytical objectives beyond those explicitly considered in this paper, the current model forms could be embedded, as appropriate, into simple simulations tied to coarsely defined events (e.g., operational days or manifested OMFs). For instance, the effects of dynamically evolving support processes readily could be played out over extended operational periods. Other analytical issues that could be addressed by similar methods are discussed below.

One simplifying assumption made consistently herein is that times to failures are governed by memoryless one-parameter exponential distributions. This is a common pragmatic approach for setting reliability requirements [2]. Alternative time-to-failure distributions could be postulated, in which case consideration would need to be given to the impact of repair/restore/replace maintenance events and the interpretation of reliability for planning and assessment purposes. In particular, different classes of recurring events may convert the 'fixed' DNMS element to 'good-as-new' or 'bad-as-old' states [5]. If the former holds universally, then the choice of the distribution is irrelevant as far as the probability of mission accomplishment calculations are concerned. Specific choices for distributions and innate parameters may, however, be of interest for tracking the demonstrated capabilities of deployed systems and projecting future performance.

Throughout the derivations, each d-day performance window implicitly is treated as probabilistically independent and identical. When $d = 5$, corresponding to a standard work week, the weekend days can be expected to offer ample time to recover before the onset of the next window. For values of $d < 5$, the lack of an early mission success in a given operational period may precipitate additional renewal efforts to prepare adequately for the advent of a follow-on performance window. If the assumption of independence cannot be defended plausibly, Markov chain methods [6] may be appropriate.

This paper's modeling framework conceptually could be expanded to incorporate explicit considerations of cost criteria encompassing design, operation, and supportability expenses, especially within the context of financial incentives associated with demonstrated operational performance of the DNMS over time. For example, contractor models relating design costs to DNMS element reliabilities can be utilized to trade off investments against possible performance-based penalties or bonuses. Similarly, the developer may determine that directly funding additional logistical support capabilities may be cost-effective in the long run.

References

[1] National Research Council, Reliability Issues for DOD Systems: Report of a Workshop, The National Academies Press, Washington, DC, 2002. (https://doi.org/10.17226/10561)
[2] National Research Council, *Reliability Growth: Enhancing Defense System Reliability,* The National Academies Press, Washington, DC, 2002. (https://doi.org/10.17226/18987)
[3] National Research Council, *Innovations in Software Engineering for Defense Systems,* The National Academies Press, Washington, DC, 2003. (https://doi.org/10.17226/10561)
[4] U.S. Department of Defense, Test and Evaluation of System Reliability, Availability, and Maintainability: A Primer, Report No. DoD 3235.1-H, Washington, DC, 1982.
[5] Beichelt, F. and Tittman, P., Reliability and Maintenance: Networks and Systems, CRC Press, Boca Raton, 2012.
[6] Privault, N., Understanding Markov Chains: Examples and Applications, Springer, Singapore, 2018.