COMBINATORIAL DESIGNS IN MULTIPLE FAULTS LOCALIZATION FOR BATTLEFIELD NETWORKS *†

Mariusz A. Fecko
Information Analysis and Services, Applied Research
Telcordia Technologies, Inc., Morristown, NJ

Małgorzata Steinder
Computer and Information Sciences Department
University of Delaware, Newark, DE

ABSTRACT

We present an application of combinatorial designs and variance analysis to correlating events in the midst of multiple network faults. Network fault model is based on the probabilistic dependency graph that accounts for the uncertainty about the state of network elements. Orthogonal arrays help reduce the exponential number of failure configurations to a small subset on which further analysis is performed. The preliminary results show that statistical analysis can pinpoint the probable causes of the observed symptoms with high accuracy and significant level of confidence. An example demonstrates how multiple soft link failures are localized in MIL-STD 188-220's Datalink layer to explain the end-to-end connectivity problems in the network layer. This technique can be utilized for the networks operating in an unreliable environment such as wireless and/or military networks.

1 INTRODUCTION

To improve the network ability to provide reliable services to end systems, a management system needs to efficiently and accurately identify the occurring network failures [13, 25]. A common procedure is to correlate network or service layer symptoms; however, this process is usually impaired by the large number of a system's layers and parameters [1, 8, 22], their interactions, and the uncertainty about their state.

This paper presents a preliminary study of applying statistical techniques [16, 18] known in the engineering quality control to cope with the exponential complexity that often hampers the event correlation process. The concept of *orthogonal arrays* [10] is used to select a feasible number of potential failure combinations. Each combination is evaluated with respect to the expected number of explained symptoms. The array's data that correspond to particular network elements are then statistically analyzed to assess with some confidence level their impact on the accuracy of symptom correlation. The elements that account for the highest variations of the correlation accuracy are selected as probable failure points.

Most of the existing techniques assume that the existence of multiple simultaneous faults is negligible [13]. Such an assumption is justified only for networks operating in a reliable environment. On the other hand, soft link failures due to jamming, misbehaving nodes, or difficult weather conditions are not likely to be limited to a single link. Similarly, battlefield or other military applications are required to function properly in an unreliable environment, where localization of multiple faults is essential. The method presented in this paper admits multiple network failures to occur at the same time. The preliminary investigation is promising—future research will further explore applications to specific military systems and protocols, e.g., MIL-STD 188-220 [7] operating in the midst of multiple link failures.

2 EVENT CORRELATION

Event correlation [11] is a commonly used technique for isolating the root cause of a problem from its reported symptoms. Efficient and precise event correlation is essential for reducing network maintenance costs and improving the availability and performance of network services.

The problems typically addressed in the literature are (1) incomplete knowledge about the existence of causal relationships between network events [13], (2) possibility of incomplete symptom observations or spurious symptoms [9, 25], (3) system adaptability to configuration changes [1], (4) ability to learn event correlation patterns [14], and (5) temporal event correlation [15]. Most of these techniques rely on the assumption that the existence of multiple simultaneous faults is negligible.

Relationships between network objects are often represented using a dependency graph [8, 12, 13]. Dependency graph is a directed acyclic graph $G(V, E)$ whose nodes V correspond to network objects (both physical and abstract) and edges E describe dependencies between the objects. Edge $(v_i, v_j) \in E$ represents the fact that object v_i affects object v_j, which we denote by $v_i \rightarrow v_j$. Every graph node may be in one of its possible states, e.g., "ok" or "not ok." In some applications it may be useful to consider other states, e.g., "transient." Nodes of dependency graph may be marked as *observation* or *failure* nodes. Some nodes are neither observation nor failure, while others may be marked as observation and failure at the same time. Intuitively, observation nodes correspond to network objects associated with observed symptoms, while failure nodes correspond to objects where unobservable root faults may happen. Let us denote by $S \subseteq V$ a set of all observation nodes, and by $P \subseteq V$ a set of all failure nodes. With every $v_i \in P$ we associate a value $pr_{i,l} = Prob\{v_i \text{ is in state } v_{i,l}\}$. This value corresponds to our prior belief that node v_i is in state $v_{i,l}$, which is independent on any symptom observations. We define function $b : E \rightarrow [0, 1]$ such that $b(v_{i,l}, v_{j,k}) = Prob\{v_j = v_{j,k} \mid v_i = v_{i,l}\}$.

We define problem as an assignment $v_i = v_{i,l}$, where $v_{i,l} \neq$ "ok." We say that problem $v_i = v_{i,l}$ explains event $v_j = v_{j,k}$ if and only if there exists a path in G from v_j to v_j. Problem $v_i = v_{i,l}$ is a root cause if there does not exist another problem $v_j = v_{j,k}$ that explains $v_i = v_{i,l}$. We say that the set of problems $\mathcal{P} = \{v_i = v_{i,l}, v_i \in P\}$ explains the set of symptoms $\mathcal{S} = \{v_j =$

*Prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

$v_{j,l}, v_j \in S\}$ if and only if for every symptom in $\mathcal{S}$ there exists its explanation in $\mathcal{P}$. We are interested in a technique that given a set of observed symptoms $\mathcal{S}$ computes the set of problems $\mathcal{P}$ such that $Prob\{\mathcal{P}|\mathcal{S}\}$ is maximum.

Event correlation may be approached as a combinatorial problem. The following optimal algorithm finds $\mathcal{P}$:

- Generate the set of all possible state assignments to nodes in P.
- For each assignment compute the probability that it explains the set of observed symptoms.
- Choose as $\mathcal{P}$ the assignment with the highest probability.

It may be observed that in the worst case the above algorithm has exponential complexity. The average computational complexity may be reduced by generating assignment combinations starting from those with the smallest number of problems [13]. This technique would be applicable if the probability of multiple simultaneous faults were very small. The approach proposed here addresses finding $\mathcal{P}$ when probability of multiple simultaneous faults is high, which is the case in, e.g., wireless and/or battlefield networks.

3 NETWORK FAULT SCENARIOS

The presented technique is applicable to correlation of any network events (in data-link, network, transport and application layers) whose causal relationships may be described using a dependency graph. The dependency graph is typically created by an expert who assigns values to all required parameters based on the system specification, history of previous failures, etc.

The example used in this paper considers link failures in data-link layer observed as a loss of end-to-end connectivity between devices in transport or application layer. Let $\mathcal{L}$ be the the set of all data-link layer links and $\mathcal{C}$ be the set of all end-to-end source-destination pairs. We define dependency graph $G(\mathcal{L} \cup \mathcal{C}, E)$, where the set of all failure nodes, $\mathcal{P} = \mathcal{L}$, and the set of all observation nodes, $\mathcal{S} = \mathcal{C}$. The set of edges in G is defined as $E = \{(p_i, c_j) \mid p_i \in \mathcal{L}, c_j = (v_k, v_l) \in \mathcal{C}\}$ and there exists a network path between v_k and v_l that includes link p_i. With every graph node $p_i \in \mathcal{P}$ we associate three values: 1 (up), 2 (down), and 3 (intermediate) indicating that link p_i is fully operational, broken, or in intermediate state (e.g., congested, out-of-sync, in transient error, etc.), respectively. The probabilities that the node assumes these values are $pr_{i,1}$, $pr_{i,2}$, and $pr_{i,3}$, where $pr_{i,1} + pr_{i,2} + pr_{i,3} = 1$. With every node in $\mathcal{S}$ two values are associated: 1 (no symptom) and 2 (symptom observed).

Function $b(p_{i,l}, c_{j,2})$ represents confidence that state $p_{i,l}$ of link p_i results in observation of connectivity problems between the source and destination denoted by c_j. The value of $b(p_{i,l}, c_{j,2})$ less than 1 indicates uncertainty if link p_i is used for communication between source and destination denoted by c_j, if such a communication has been attempted, or if link failure type is severe enough to cause the high level problem.

To perform root cause analysis of connectivity problems between nodes A and B one needs to know the path over which packets between A and B are routed. This information is available in routing or data-link layer forwarding tables. However, due to automatic system reconfiguration and high overhead associated with collecting routing tables, it may not be possible to make the most up-to-date routing information available to the correlation process. To represent possible causes of connectivity problems between nodes A and B the graph should include all links that could be used for transferring packets between A and B. Function b should differentiate the links with respect to their likelihood of having been utilized in the communication between the two nodes.

Figure 1 presents dependency graph for a 5-node source routing wireless network. While in general building a dependency graph for wireless networks is difficult, source routing makes it possible to determine which links could have been used for communication between two nodes when this communication fails. Since the information is readily available to both communicating nodes, it may be provided to the correlation system along with the failure symptom. In addition, it is unnecessary to build the complete dependency graph in advance—it may be dynamically extended by adding observation nodes only when corresponding symptoms arrive. Some source routing protocols (e.g., MIL-STD 188-220's Source Directed Relay [7]) find not only the best route between two nodes A and B but also include several alternative routes in a routing tree. All links included in any of the routes are presented as possible causes of connectivity problems between A and B. The values assumed by function b for links included in the primary route (appear in bold) are greater from the values assumed by function b for links not included in the primary route. Table 1 presents dependency graph parameters for the example in Figure 1.

	$p_{i,l}$		
	up	*intermediate*	*down*
primary-route link	0	0.7	1.0
not a primary-route link	0	0.1	0.3

Table 1: Values of $b(p_{i,l}, c_{j,2})$ for the dependencies in Figure 1.

4 COMBINATORIAL DESIGNS PARADIGM

Recall from Section 3 that the network environment considered here is described by a set of links, each being in up, down, or intermediate state at a given point in time. In a wireless intranet consisting of n nodes, there are up to $n(n-1)/2$ links, which leads to an enumeration of $3^{n(n-1)/2}$ possible interconnection configurations. This makes it infeasible to apply any kind of analysis based on the entire set of configurations as even for small values of n, the number of configurations becomes prohibitively large. For example, in MIL-STD 188-220B [7] Intranet Layer, there may be as many as $n = 16$ nodes yielding up to $3^{120} = 1.8e57$ configurations.

To cope with the above problem, we will adopt the combinatorial design paradigm [4, 6, 10, 16], which has been successfully used in applications ranging from medicine and biology, to quality engineering [18], to testing network interfaces [24, 23] and software intensive systems [3, 4, 5].

4.1 Statistical Paradigm

Suppose that a system is described by parameters $p_1, \ldots, p_k$, where each p_i assumes q_i values $p_{i,1}, \ldots, p_{i,q_i}$. In the statistical terminology, we say that each *factor* p_i has a *level* of q_i [6, 10]. The number of configurations is therefore equal to $\prod_{i=1}^{k} q_i$. A

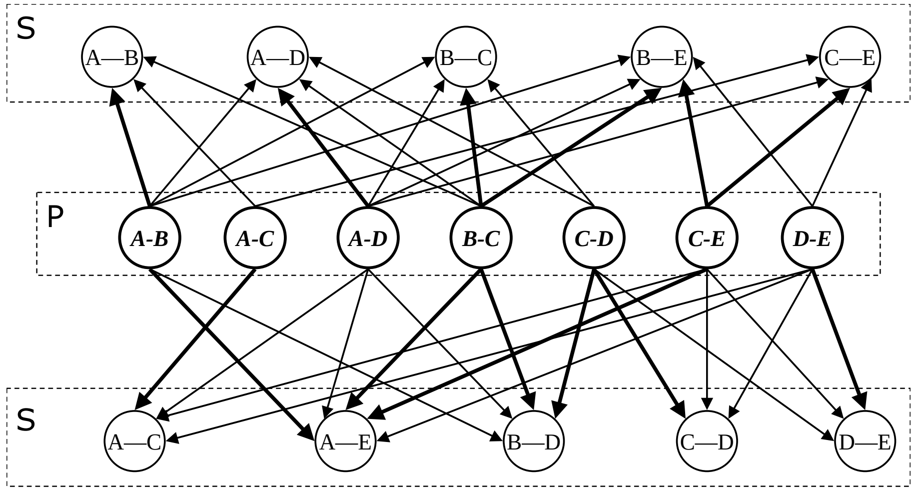

Figure 1: Dependency graph for a sample 5-node network. Primary links appear in bold.

variation of the combinatorial design paradigm, called *factor-covering designs* [6], generates configurations that cover all m-way interactions of factors such that for any $m \leq k$ factors $p_{i_1}, \ldots, p_{i_m}$ and any value p_{i_j, l_j} for each p_{i_j}, there exists a configuration with the values of $p_{i_1,l_1}, \ldots, p_{i_m,l_m}$. For example, for $k = 10$, where each p_i assumes binary values of 0 and 1, there are 2^{10} possible configurations. However, all 2-way interactions can optimally be covered by only six configurations of (0000000000), (0000111111), (0111000111), (1011011001), (1101101010), and (1110110100) [6]. It can be shown [3] that, for fixed m, the number of configurations in factor-covering designs grows logarithmically in the number of factors k.

If a design satisfies an additional requirement that each m-way interaction $p_{i_1,l_1}, \ldots, p_{i_m,l_m}$ appears the same number of times in the generated set of configurations, it is called an *orthogonal array (OA)* [10, 17]. An OA with N rows, k columns, and each $q_i = q$, is denoted as $OA(N, q^k)$. Orthogonal arrays are the statistical basis for the Taguchi methods [18] widely used in engineering quality control. The penalty for having a balanced set of m-way interactions is a larger set of configurations, which no longer grows logarithmically. For example, some of the standard OAs are $OA(k+1, 2^k)$ for $m = 2$ and $OA(2k, 2^k)$ for $m = 3$.

4.2 Applications

In the application to software testing [3, 4, 5], the software under test is described by a set of test parameters (user inputs, external events, fields on the screen) such that each interaction of test parameters' values gives a different test configuration. The combinatorial designs offer a methodology for selecting a relatively small number of test configurations from the infeasibly large number of possible ones. This approach is used in several commercial products such as Bell Labs CATS system [19] and Telcordia AETG system [3]. The AETG system implements a variety of configuration generating algorithms, and has been applied for example to the testing of the Integrated Services Control Point [5].

In the network environment considered in this paper, the number of interconnections used to compute the model probabilities is reduced by applying a combinatorial design with $k \leq n(n-1)/2$ factors. The i-th link is described by factor p_i with values $p_{i,1} = 1$, $p_{i,2} = 2$, and $p_{i,3} = 3$ (all levels are $q_i = q = 3$). Since our event correlation technique makes its decisions based on statistical analysis of the entire set of configurations, orthogonal arrays are the preferred one of the two types of designs.

In the applications of the paradigm reported in the computing literature [3, 5, 23], the values of $m \geq 4$ are rarely used. Typically, pairwise ($m = 2$) or triple ($m = 3$) interaction coverage is considered sufficient. Empirical studies show that very few additional system faults can be revealed by increasing m beyond three, at the expense of the undesirable increase of the number of configurations. Therefore, the network management system considered in this paper should balance its computing power, the desired fault identification delay, and the number of network nodes to adaptively choose either pairwise or triple interactions. The technique presented here uses OAs with each column assigned to a single factor (link), which may confound the effects of a single link with those of link interactions. Another possibility is to use larger OAs with additional columns assigned to factor interactions, but such designs add significant computing complexity.

5 COMBINATORIAL DESIGNS IN EVENT CORRELATION

An event correlation algorithm evaluates each interconnection configuration with respect to its ability to explain the set of nodes where symptoms $\mathcal{S}_{\text{obs}} \subseteq \mathcal{S}$ have been observed. In this process, each r-th row of the correlation OA is assigned ranking y_r that estimates the expected value of the number of explained symptoms:

$$y_r = \sum_{c_j \in \mathcal{S}_{\text{obs}}} (1 - \prod_{p_i \in \mathcal{P}} (1 - b(p_{i,l_r}, c_{j,2}))) \quad (1)$$

The computation in (1) can be done in advance in $O(|\mathcal{S}| * k^2)$ time. Then, the real-time event correlation adopts the well-known statistical techniques [16, 18] to determine factors (links) whose potential failures account for the greatest variations in y across all rows of the OA. The entire real-time computation can be performed in $O(kN) = O(k^2)$ time.

A statistical technique used here is a simple ANOVA [16, 18] performed for each column in the OA. Let us introduce the following notation:

- $y_{i,l}$—value of y under level $p_{i,l}$
- $Y_{i,l}$—sum of y under level $p_{i,l}$
- $n_{i,l}$—number of rows with level $p_{i,l}$ (all $n_{i,l} = n$)
- $\bar{Y}_{i,l}$—average of y under level $p_{i,l}$
- T—sum of y across all rows
- $\bar{T}$—average of y across all rows = T/N

The total variability SS_T of ranking y can be partitioned into the links' main effects (SS_i for i-th link) and the residual effect. The residual SS_{res} contains the estimate of the effects caused by the link interactions. Let us first compute the sum of squares of the differences between the effects of the link levels and the grand average as follows:

$$SS_i = n * \sum_{l=1}^{q} (\bar{Y}_{i,l} - \bar{T})^2 \qquad (2)$$

The residual is computed by subtracting the cumulative links' main effect from the total variability measure:

$$SS_{\text{res}} = SS_T - \sum_{i=1}^{k} SS_i \qquad (3)$$

The degrees of freedom for SS_i and SS_{res} are equal to $\nu_1 = q-1$ and $\nu_2 = (N-1) - k*(q-1)$, respectively. For each link i, the value of $MS_i = SS_i/\nu_1$ estimates the individual variance derived from the variance of the sample averages $\bar{Y}_{i,1}, \ldots, \bar{Y}_{i,q}$. To evaluate the link main effect, MS_i is compared with the residual mean square $MS_{\text{res}} = SS_{\text{res}}/\nu_2$ using the standard F-test [16, 18], where $F_{\text{data}} = MS_i/MS_{\text{res}}$. The value of F can also be obtained from the look up tables [18] as F_{α,ν_1,ν_2} based on the desired significance level α. If $F_{\text{data}} \geq F_{\alpha,\nu_1,\nu_2}$, we can claim with confidence $(1-\alpha)$ that factor p_i affects ranking y. In this case, the level of p_i that corresponds to a high value of y is the statistically estimated state of i-th link. If $F_{\text{data}} < F_{\alpha,\nu_1,\nu_2}$, we do not have enough confidence to draw any conclusion about the state of p_i from the set of observed symptoms.

It is worth pointing out that the technique presented here does not currently attempt to produce a set of problems $\mathcal{P}$ explaining all observed symptoms $\mathcal{S}_{\text{obs}}$. Instead, each failure node p_i passing F-test is assigned state $p_{i,l}$ that is believed to have contributed to observing symptoms in $\mathcal{S}_{\text{obs}}$.

The current ranking y in (1) favors faults that are able to explain the most observed symptoms. The faults belonging to the first formulated fault hypothesis may be unable to explain all the observed symptoms. A separate run of the algorithm is necessary to find the root cause of the symptoms not explained by the first fault hypothesis. While this approach seems reasonable, we believe that further research should bring about more versatile ranking scheme that allows to find explanation for all the observed symptoms in the first iteration. Such scheme could take into account not only how well a symptom is explained by a given configuration, but also how likely it is to find this symptom's explanation in other configurations. Another scheme can rank configurations according to the likelihood that at least one symptom is left unexplained. This would put more weight on the faults that can explain few symptoms, but are the only ones to do so.

links	configuration	y	links	configuration	y
1	3333333333	3.468	2	1311122312	1.470
3	2322211321	4.000	4	3131112231	3.460
5	1112231213	2.610	6	2123323222	4.000
7	3232221132	3.679	8	1213313111	1.800
9	2221132123	3.510	10	3313111223	3.370
11	1321233232	3.868	12	2332322211	3.058
13	3111223121	3.370	14	1122312133	3.544
15	2133131112	2.530	16	3212332322	3.190
17	1223121331	3.433	18	2231213313	3.301
19	3323222113	3.433	20	1331311122	3.460
21	2312133131	2.230	22	3121331311	3.180
23	1132123323	3.460	24	2113212332	3.349
25	3222113212	3.510	26	1233232221	3.853
27	2211321233	3.370			

Table 2: Orthogonal array for 10 links—columns correspond to links (A-B), (A-C), (A-D), (A-E), (B-C), (B-D), (B-E), (C-D), (C-E), and (D-E).

In the MIL-STD 188-220 example with 10 links, we use an $OA(27, 3^{13})$ [10] shown in Table 2, where the last three factors are removed, $N = 27$, $n = 9$, $q = 3$, $\nu_1 = 2$, $\nu_2 = 6$, and each 2-way interaction appears three times.

Suppose that the set of observed symptoms is $\mathcal{S}_{\text{obs}} = \{(A \leftrightarrow D), (C \leftrightarrow E), (A \leftrightarrow E), (B \leftrightarrow D)\}$. The minimum acceptable level of confidence is assumed 95%, with the minimum $F_{0.05,2,6} = 5.14$. To perform the analysis of variance, we used Splus—an interactive environment for data analysis and graphics [2]. As shown in Table 3, we can claim with at least 95% confidence (i.e., our claim may be false with 5% risk) that changes in the state of links (A-D), (B-C), (C-E), (A-B), and (C-D) produce the given symptoms. In fact, the values of F_{data} for links (A-D), (B-C), and (C-E) allow us to make this claim with much higher confidence 99%. The faults in these three links explain all the symptoms in $\mathcal{S}_{\text{obs}}$.

A problem with the above analysis is that single links' effects are confounded with the link interactions (2-way, 3-way, and higher), which may sometimes lead to incorrect conclusions. For example, in Table 3, link (A-B)'s effect is significant at 5%, whereas the high value of F may in some cases represent the effect of other links' interactions that are confounded with (A-B). Additional statistical analysis is needed to obtain more reliable results [16].

To assign specific states to the three significant links, we first notice that the values of $\bar{Y}_{i,l}$ are higher for $l = 2, 3$ than for $l = 1$. The desired level of confidence that pairwise differences in $\bar{Y}_{i,l}$ are meaningful for particular states of the link may be verified by performing additional statistical tests (e.g., comparison of level means [16]). In practice, we are primarily interested in (1) selecting links that caused the observed symptoms, which should be determined with high confidence $(1-\alpha)$), and (2) gaining some confidence that the states other than $l = 1$ (link up) have caused

	link	F_{data}	*selected*	*confidence*
1	(A-B)	5.48	yes	≥95%
2	(A-C)	0.91	no	n/a
3	(A-D)	33.10	yes	≥99%
4	(A-E)	0.08	no	n/a
5	(B-C)	10.40	yes	≥99%
6	(B-D)	0.86	no	n/a
7	(B-E)	0.33	no	n/a
8	(C-D)	6.58	yes	≥95%
9	(C-E)	31.59	yes	≥99%
10	(D-E)	1.38	no	n/a

Table 3: The F-test results for the ten links.

the symptoms. The latter confidence may possibly be lower than $(1-\alpha)$, since after pinpointing selected links, the management system usually performs testing. For example, for link (A-D), the mean values are $\bar{Y}_{3,1}=2.71$, $\bar{Y}_{3,2}=3.60$, and $\bar{Y}_{3,3}=3.31$, pointing to states $p_{3,2}$ and $p_{3,3}$, i.e., link (A-D) down or in intermediate state, as one of the likely causes of the observed symptoms.

6 CONCLUSIONS AND FUTURE WORK

This paper presents an application of combinatorial designs and variance analysis to correlating events in the midst of multiple network faults. Network fault model is based on the probabilistic dependency graph that accounts for the uncertainty about the state of network elements. Orthogonal arrays help reduce the exponential number of failure configurations to a small subset on which further analysis is performed.

Among the advantages of the proposed technique one can list the ability to define various ranking schemes suitable for different objectives of the network management system. In this paper we used a simple ranking based on the expected number of the explained symptoms. The ranking scheme could be defined so that the resultant fault hypothesis should first propose faults that are easier to test and/or repair. Alternatively, the preference could be established based on fault severity.

The preliminary results show that statistical analysis pinpoints the probable causes of the observed symptoms with high accuracy and significant level of confidence. A case study demonstrates how multiple soft link failures are localized in MIL-STD 188-220's Datalink layer to explain the end-to-end connectivity problems in the network layer.

This technique can be utilized for the networks operating in an unreliable environment such as wireless and/or military networks. Further research will investigate more advanced combinatorial designs and statistical analysis techniques, explanation of symptoms caused by interaction of multiple faults, building the relevant orthogonal arrays from the past fault history in the case where a dependency graph is not given, and applications to specific military systems and protocols.

Current research at the University of Delaware extensively investigates other probabilistic approaches to event correlation. Recently, several algorithms have been designed and analyzed including the most probable explanation in belief networks and iterative hypothesis update [21]. [1]

[1] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Lab or the U.S. Government.

References

[1] K. Appleby, G. Goldszmidt, and M. Steinder. Yemanja—a layered event correlation engine for multi-domain server farms. In *IFIP/IEEE Intg'ed Network Management VII*, Seattle, WA, 2001.

[2] R. A. Becker and J. M. Chambers. *S—An Interactive Environment for Data Analysis and Graphics*. Statistics/Probability Series. Wadsworth Pub Co, Belmont, CA, 1984.

[3] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton. The AETG system: An approach to testing based on combinatorial design. *IEEE Trans. Softw. Eng.*, 23(7), 1997.

[4] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, pp. 83–87, 1997.

[5] S. R. Dalal, A. Jain, N. Karunanithi, J. M. Leaton, and C. M. Lott. Model-based testing of a highly programmable system. In *Proc. IEEE ISSRE: Int'l Symp. Softw. Reliability Eng.*, Paderborn, Germany, 1998.

[6] S. R. Dalal and C. L. Mallows. Factor-covering designs for testing software. *ASQ/ASA Technometrics*, 40(3), 1998.

[7] DoD. *Military Standard—Interoperability Standard for Digital Message Device Subsystems (MIL-STD 188-220B)*, 1998.

[8] R. Gopal. Layered model for supporting fault isolation and recovery. In *Proc. IEEE/IFIP Network Operation and Management Symp.*, Honolulu, Hawaii, 2000.

[9] M. Hasan, B. Sugla, and R. Viswanathan. A conceptual framework for network management event correlation and filtering systems. In Sloman et al. [20], pp. 233–246.

[10] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal Arrays: Theory and Applications*. Springer Series in Statistics. Springer-Verlag, New York, NY, 1999.

[11] G. Jakobson and M. D. Weissman. Alarm correlation. *IEEE Network*, pp. 52–59, 1993.

[12] S. Kätker and K. Geihs. A generic model for fault isolation in integrated management systems. *J. Network and Syst. Management*, 5(2):109–130, 1997.

[13] I. Katzela and M. Schwartz. Schemes for fault identification in communication networks. *IEEE Trans. Networking*, 3(6), 1995.

[14] L. Lewis. A case-based reasoning approach to the resolution of faults in communications networks. In *IFIP/IEEE Intg'ed Network Management III*, pp. 671–681, San Francisco, CA, 1993. North-Holland.

[15] G. Liu, A. K. Mok, and E. J. Yang. Composite events for network event correlation. In Sloman et al. [20], pp. 247–260.

[16] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley & Sons, New York, NY, 5th ed., 2000.

[17] C. R. Rao. Factorial experiments derivable from combinatorial arrangements. In *A Survey of Combinatorial Theory*. North-Holland, 1973.

[18] P. Ross. *Taguchi Techniques for Quality Engineering*. McGraw-Hill, New York, NY, 1988.

[19] G. B. Sherwood. Effective testing of factor combinations. In *Proc. STAR: Int'l Conf. Softw. Testing Analysis Review*,

Jacksonville, FL, 1994.

[20] M. Sloman, S. Mazumdar, and E. Lupu, eds. *IFIP/IEEE Intg'ed Network Management VI*, Boston, MA, 1999.

[21] M. Steinder and A. S. Sethi. Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In *Proc. IEEE IC3N: Int'l Conf. Comp. Comm. Networks*, Scottsdale, AZ, 2001.

[22] M. Steinder and A. S. Sethi. The present and future of event correlation: A need for end-to-end service fault localization. In *Proc. IIIS SCI: World Multi-Conf. Systemics Cybernetics Informatics*, Orlando, FL, 2001.

[23] A. L. Williams and R. L. Probert. A practical strategy for testing pairwise coverage at network interfaces. In *Proc. IEEE ISSRE: Int'l Symp. Softw. Reliability Eng.*, White Plains, NY, 1996.

[24] A. W. Williams. Determination of test configurations for pair-wise interaction coverage. In *Proc. IFIP TestCom: Int'l Conf. Test. Communicat. Syst.*, pp. 59–74, Ottawa, Canada, 2000. Kluwer Acad. Pub.

[25] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High speed and robust event correlation. *IEEE Communications*, 34(5):82–90, 1996.