Calculation of Node-Pair Reliability in Large Networks with Unreliable Nodes

Don Torrieri, Senior Member IEEE
Department of the Army, Adelphi

Key Words — Node-pair reliability, unreliable node, network reliability

Reader Aids —
General purpose: Extend the state of the art
Special math needed for explanations: Set theory and probability
Special math needed to use results: Same
Results useful to: Network designers and reliability analysts

Summary & Conclusions — A new efficient method that compensates for unreliable nodes in network reliability computations is presented. This method can be embedded in the modified Dotson algorithm or any algorithm that generates a symbolic reliability expression for networks with perfect nodes. Its cost increases linearly with the number of links, and the effect of unreliable nodes can be directly computed. This method supplants the Aggarwal method and other methods of compensating for unreliable nodes in the calculation of node-pair reliability. When combined with the modified Dotson algorithm, this method provides accurate reliability estimates for networks so large that the Theologou-Carlier algorithm cannot complete its computations in a reasonable amount of time. For such networks, the new method can be embedded in the modified Dotson algorithm to estimate both the node-pair reliability and the error in this estimate even if the algorithm is terminated before completion.

1. INTRODUCTION

The node-pair (2-terminal) reliability is the probability that at least one path exists between a source node and a terminal node in a directed network. The node-pair reliability is an important performance measure in communication networks that use flooding for route setup or packet transmission. The probability of successful flooding from a source node to a terminal node is equal to the node-pair reliability.

To account for node failures, the best and most commonly used method is that of Aggarwal, Gupta, Misra (AGM) [1]. This method can be embedded in any algorithm that generates a symbolic reliability expression for networks with perfect nodes. The AGM method expands each term of the reliability expression derived for perfect nodes and replaces the variables by functions of node & link variables. After this substitution, Boolean simplification might be needed. Unfortunately, the cost of these operations can rise exponentially with the number of links. Furthermore, the use of symbolic calculations rather than direct numerical ones can require prohibitively large storage [2].

The new method (NPR/T') is much simpler, more direct, and more rigorously derived than AGM, and can be embedded in the same algorithms. The cost of this new compensation method increases linearly with the number of links, and the effect of the unreliable nodes can be directly computed.

NPR/T can be embedded in the modified Dotson algorithm [3], which is one of the most computationally efficient of the many algorithms to calculate the node-pair reliability of networks with completely reliable nodes [3,4]. This algorithm continually calculates both upper & lower bounds on the node-pair reliability until completion of the algorithm. These bounds can be used to estimate both the node-pair reliability and the error in this estimate even if the algorithm is terminated before completion.

The modified Dotson algorithm with the embedded NPR/T is called the combined algorithm.

The algorithm of Theologou & Carlier (TC) [2], which uses factoring & reductions, rapidly computes node-pair reliability in a small network with unreliable nodes. However, in its present form, TC has 3 difficulties:

1. TC is useful only if run to completion, which is impractical for large networks (exceeding 100 nodes + links). No modification that allows this algorithm to track the upper & lower bounds has yet been published.
2. The computation must be repeated if any of the node or link reliabilities changes. In contrast, the combined algorithm can store success & failure events and recompute the node-pair reliability without running the entire algorithm again.
3. TC does not allow specifying the maximum number of links in a path. Such a maximum exists in most practical communication networks and can easily be specified in the combined algorithm.

Notation
s, t [source, terminal] nodes of node pair
n, m number of [nodes (vertices), links (edges)] in the network
αi, αs, βi Pr{[node i, source node s, link i] of network is operational}
S, F, \( |S|, |F| \) number of [success, failure] events
Sl, Fj event: links with terminal node j are [operational, failed] as specified by \([S, F]\)
R node-pair reliability for s & t
Nj number of [failed, operational] links directed into node j

Editors’ note: We have assigned this acronym NPR/T (node-pair reliability - Torrieri) for simple, clear, unique reference to the concept.

US Government work not protected by US copyright.
2. NPWT (COMPENSATION) METHOD

Assumptions

1. All node & link failures are mutually s-independent of each other.
2. The compensation method is embedded in a node-pair reliability algorithm.
3. Each node, link, group, and the network is either operational or failed.
4a. A node is operational iff it functions as intended.
4b. A link is operational iff it allows communication from its initial node to its terminal node.

The event [successful communication over a link] = the event [both the link and its terminal node are operational]. To exploit this relation, a network with unreliable nodes is replaced by an equivalent network with the same probabilities that communication can occur over the links but with completely reliable nodes except for the source node. The original node reliabilities are subsumed into the link reliabilities. Consider link i and its terminal node; the link i in the equivalent network has reliability $\beta_i \cdot \alpha_i = \Pr\{\text{successful communication occurs over the link}\}$.

In the equivalent network, the link failures are not necessarily mutually s-independent, but a link still fails s-independently of another link with a different terminal node. $S_j$ specifies certain operational links and certain failed links in a network realization. Group the links of both types that have the same terminal node; then:

$$\Pr\{S_j\} = \alpha_j \cdot \prod_{i=1}^{n-1} \Pr\{S_{ij}\}.$$  \hspace{1cm} (1)

Let $S_j$ not specify the status of any links directed into node j, then $\Pr\{S_{ij}\} = 1$.

Let $S_j$ specify that links 1, 2, ..., $K_j$ directed into node j are operational, and not specify the status of other links directed into node j, then,

$$\Pr\{S_{ij}\} = \alpha_j \cdot \prod_{i=1}^{K_j} \beta_i.$$  \hspace{1cm} (2a)

because these links are not operational in the equivalent network unless all these links and node j are operational in the original network.

Let links 1, 2, ..., $N_j$ directed into node j be specified as failed and links $N_j+1, N_j+2, ..., N_j+K_j$ be specified as operational, and let $K_j \geq 1$. No links can be operational in the equivalent network unless node j is operational in the original network. Given that node j is operational, the links are operational or have failed in the equivalent network depending upon whether they have failed in the original network. Because of the s-independence of link failures in the original network, it follows that, for any $N_j \geq 0$:

$$\Pr\{S_{ij}\} = \alpha_j \cdot \prod_{i=1}^{N_j} (1-\beta_i) \cdot \prod_{i=N_j}^{N_j+K_j} \beta_i,$$  \hspace{1cm} (2b)

Let $K_j = 0$, then links 1, 2, ..., $N_j$ have failed in the equivalent network iff [node j is failed] or [all $N_j$ links are failed and node j is operational]. Thus,

$$\Pr\{S_{ij}\} = 1 - \alpha_j \cdot \prod_{i=1}^{N_j} (1-\beta_i),$$  \hspace{1cm} (2c)

Because the $S_j$ are disjoint (mutually exclusive events), the node-pair reliability is the sum of the probabilities of all success events:

$$R = \sum_{i=1}^{[S]} \Pr\{S_i\}. \hspace{1cm} (3)$$

The combined method is implemented by using (1) & (2) in (3).

By analogy with (1) & (2),

$$\Pr\{F_i\} = \alpha_i \cdot \prod_{k=1}^{n-1} \Pr\{F_{ik}\};$$  \hspace{1cm} (4)

$$\Pr\{F_{ij}\} = 1 - \alpha_j \cdot \prod_{i=1}^{N_j} (1-\beta_i),$$  \hspace{1cm} (5)

3. COMBINED METHOD

NPWT is embedded in the modified Dotson algorithm to give the combined method. If the algorithm execution is stopped before it is completed, then:

$$M_F \cdot \sum_{i=1}^{M_S} \Pr\{S_i\} \leq R \leq \alpha_s - \sum_{i=1}^{M_F} \Pr\{F_i\} \hspace{1cm} (6)$$

Notation

$M_S, M_F$ size of incomplete [success, failure] collection.

The bounds provided by (6) become tight rapidly because the algorithm tends to examine success & failure events in descending order of their size, where size is determined by the number of elementary events contained in an event.

If the node & link probabilities are fixed and the upper & lower bounds in (6) are calculated iteratively as the success & failure events are found, then it is not necessary to store success & failure collections. It is desirable to store success & failure collections when one wants to compute the node-pair...
reliability as a function of various values of \( \alpha_i, \beta_i \). This storage obviates the regeneration of the success & failure events whenever the node & link probabilities change. Any change merely requires the recalculation of (3), (5), or (6).

Initial pruning of the network representation can sometimes expedite the execution of the algorithm. The pruning suggested by Page & Perry [4] can easily be modified to apply to a network with unreliable nodes.

4. COMPARISON WITH THEOLOGOU-CARLIER ALGORITHM

The Theologou-Carlier algorithm (TC) is faster than the combined method when:
- both algorithms are run to completion, and
- the network is small (50 or fewer nodes + links, as in the examples in [2]).

However, since the combined algorithm can be truncated and still produce tight bounds, it is much faster for large networks.

4.1 Example 1

Assumptions:

- \( m = 80 \) links, \( n = 25 \) nodes
- \( \beta_i = 0.90, i = 1, 2, \ldots, m \)
- \( \alpha_i = 0.99, i = 1, 2, \ldots, n \).

The combined method was compared with TC by computing the corner-to-corner reliability of the network in figure 1. The computations were done on an Avalon 386 computer with a 120-MB hard-disk drive, and the programming language was PASCAL. The combined method, although unable to run to completion within 30 minutes, produced bounds separated by less than 0.01 in 41 seconds. Thus, a reliability estimate equal to the average of these bounds has an error no larger than 0.005, which is quite acceptable in practice. TC, which must be run to completion, computed the reliability in 30 minutes.

4.2 Example 2

\( m = 128 \) links, \( n = 34 \) nodes

The node-pair reliability was computed [5] for nodes 20 & 25 in the network of figure 2. The combined method produced an estimation error less than 0.005 in 16 seconds. TC did not complete its computation in 14 hours.

ACKNOWLEDGMENT

I am pleased to acknowledge the contribution of J.S. Lee Associates of Rockville who carefully programmed the algorithms in this paper and obtained the numerical results.

REFERENCES


AUTHOR

Dr. Don Torrieri; AMSRL-SS-IB; Dept. of the Army; 2800 Powder Mill Road; Adelphi, Maryland 20783-1145 USA.
Internet (e-mail): dtorr@arl.mil (Continued on page 382)
6. DISCUSSION

The algorithm can be made more efficient easily by recognizing the fact that during the computation, once a zero capacity is encountered, there is no need to proceed any further for that ΔP. This can save much computation. For example, while computing PI for the networks in examples 1 & 2, 55% & 64% of the computations for ΔPI can be saved. The computation times for examples 1 & 2 were 0.05 & 0.11 sec respectively.

REFERENCES


AUTHORS

Dr. Pramod K. Varshney; Electrical & Computer Eng’g Dept; Syracuse University, Syracuse, New York 13244 USA.

Pramod K. Varshney (S’72,M’77,SM’82) was born in 1952. He received the BS (1972) in Electrical Engineering and Computer Science (with highest honors), and the MS (1974) and PhD (1976) in Electrical Engineering from the University of Illinois at Urbana-Champaign. Since 1976 he has been with the Electrical & Computer Engineering Department at Syracuse University, Syracuse where he is a Professor and the Associate Chair’n of the department.

Abhay R. Joshi; Codex Corp; 20 Cabot Blvd; Mansfield, Massachusetts 02048 USA.

Abhay R. Joshi was born in 1966 in Dhubi, India. He received his BTech (1988) from the Indian Institute of Technology, Bombay and his MS (1990) from Syracuse University, Syracuse. He is employed at Codex Corp. and is working on computer networks.

Pei-Liang Chang; Electrical & Computer Eng’g Dept; Syracuse University; Syracuse, New York 13244 USA.

Pei-Liang Chang was born in Taiwan, Rep. of China, in 1956. He received his BS (1978) in Electrical Engineering from Chen-Kung University, Taiwan and MS (1985) in Computer Engineering from Syracuse University, Syracuse.

He has been with Chung-Shan Institute of Science & Technology, Taiwan since 1980 and is on leave at Syracuse University working toward his PhD.

Manuscript received 1994 April 28.

IEEE Log Number 94-04552

Calculation of Node-Pair Reliability in Large Networks with Unreliable Nodes

(Continued from page 377)

Don Torrieri (SM’90) received a BS from the Massachusetts Institute of Technology, MS from the Polytechnic University, and PhD from the University of Maryland. He analyzed electronic systems at the Naval Research Laboratory. Since 1977, he has been employed by the Department of the Army. He is a Fellow of the Army Research Laboratory. His primary interests are the analysis & assessment of adaptive arrays, communication networks & systems, and neural networks. He is the author of Principles of Secure Communication Systems (2nd ed) 1992, Artech House; Principles of Military Communication Systems 1981, Artech House; chapter 1 of Acousto-Optic Signal Processing 1983 (1994, 2nd ed), Marcel Dekker, and many journal articles. He received the Best Paper award of the 1991 IEEE Military Communications Conference.

Manuscript received 1993 July 23.

IEEE Log Number 92-15601