AN ALGORITHM FOR THE COMPUTATION OF COMPUTER NETWORK RELIABILITY IN LINEAR-TIME

Steve G. Belovich  Vijaya K. Konangi
Department of Electrical Engineering
Cleveland State University
Cleveland, Ohio  44115

ABSTRACT

The best known solution methods for network reliability problems are of exponential time-complexity. This exponential behavior can render even moderately sized problems computationally intractable due to the enormous amount of time required to generate a solution. To render such problems manageable, a bounding algorithm has been developed that provides upper and lower bounds for the source-to-terminal reliability of an arbitrary network. A unique feature of this bounding algorithm is that it possesses linear time-complexity when the maximum indegree of all network nodes is limited.

INTRODUCTION

Network reliability is usually expressed as the probability that a contiguous path exists between a source node and a destination node, under various assumptions regarding the statistical behavior of the allowable failures. More general forms of this measure, called the source-to-terminal (s-t) reliability, have also been proposed such as K-terminal reliability, which is the probability that all nodes in a given set K can successfully communicate. Many methods have been described in the literature for the computation of these and related quantities [1,3,8,9,10,11], but all such methods are computationally intractable for networks of even moderate size. This intractability is due to the exponential relationship between the solution time and the network size, and can occur despite sundry simplifying assumptions. It has been shown that the determination of the K-terminal reliability for an arbitrary graph is, in fact, NP-hard [7,13]. The best known solution techniques for NP-hard problems are of exponential order, which can render network reliability problems of practical importance unsolvable.

Despite the computational difficulties, reliability analyses of practical networks are still necessary and greatly desired. The approach taken here is to trade accuracy for efficiency by judicious invocation of several approximations. In particular, the reliability of a given node may be recursively written in terms of the reliabilities of those nodes which have links incident into it. This yields an equation with a conditional probability term which is amenable to a rapid approximate evaluation. If statistical independence is assumed during the computation of this conditional probability term, an upper bound to the exact s-t reliability is obtained. A different approximation in the computation of this conditional probability term yields a lower bound. These approximations allow the estimation of the conditional probability term within linear-time. This procedure thus yields a linear-time algorithm that produces both upper and lower bounds for the s-t reliabilities of all nodes in a given network. These s-t reliabilities are defined with respect to an arbitrary starting (or "root") node and represent the probabilities of reaching each of the network nodes from the root node.

DEFINITIONS AND NOTATION

Root Node: The root node of a network graph is that node from which the s-t reliability is to be computed.

Source Node: A source node S of a given network vertex V is a node that has at least one link, from itself, incident into vertex V.

Simple Path: A series combination of links by which a given vertex i may be reached from another vertex j such that no link is traversed more than once.

Operational Source Node: A source node is termed operational if it is connected to the root node by at least one simple path.

R(V): The reliability of network vertex V, that is, the probability that vertex V is connected to vertex 1 by at least one simple path.

1;: The event that link i is operational.

S;: The event that source node i of vertex V is connected to the root node by at least one simple path.

U: Set union.
THE METHOD

General Approach to s-t Reliability Analysis

The reliability analysis technique to be derived here assumes that the network links are the only failure-prone components and that failures occur independently. Consider the network $G = (V,E)$ where $V$ corresponds to the set of vertices and $E$ to the set of edges. Let an arbitrary vertex $V$ have $n$ source nodes. Let $S_1, \ldots, S_n$ denote the events that source nodes $1, \ldots, n$ of vertex $V$ are operational, that is, they are connected to vertex 1 (the root node) by at least one simple path, respectively. Let $I_1, \ldots, I_n$ denote the events that links 1, $i = 1, \ldots, n$, which join the $n$ source nodes of $V$ to vertex $V$ are operational, respectively.

The reliability of vertex $V$ is

$$R(V) = P(S_1, \ldots, S_n) = P(S_1, I_1, \ldots, I_n)$$

With the definition of $R_{old}(V)$ as the compound event,

$$R_{old}(V) = S_1, I_1, \ldots, S_n, I_n$$

the node reliability $R(V)$ becomes

$$R(V) = P(R_{old}(V)) + P(S_1, I_1)$$

$$- P(S_1, I_1) P(R_{old}(V)) P(R_{old}(V))$$

The above expression, which is true for $1 \leq n \leq \text{indeg}(V)$, has several computational advantages. First, the contribution of each source node to the total reliability of vertex $V$ is computed sequentially and thus the exponential time complexity associated with a full expansion of equation (1) is reduced by this approach to linear [2]. Second, the probabilistic interdependencies of the event-pairs $S_i, I_i$ that arise from interconnections among the source nodes (and any other nodes) are implicitly accounted for by the conditional probability $P(S_1, I_1 | R_{old}(V))$. It should also be noted that equation (3), like equation (1), is exact, for if all the indicated quantities could be precisely evaluated, then the reliability of vertex $V$, $R(V)$, would be exact.

The major drawback associated with using equation (3) to compute the reliability $R(V)$ is that the conditional probabilities are difficult to compute in practice due to the enormous number of event occurrence possibilities that must be generated. The tightest bound on the number of such possibilities is exponential and thus computational intractability is not entirely eliminated by equation (3). It should be emphasized, however, that this is not always true and that equation (3) can yield exact vertex reliabilities for certain simple classes of networks in linear time.

An Upper Bound

Unfortunately, for network topologies of practical interest, equation (3) alone is insufficient to yield exact answers in linear time. If, however, it is assumed that the events which define the conditional probability occur independently, then

$$P(S_1, I_1 | R_{old}(V)) = P(S_1, I_1)$$

Applying this approximation to the recursive formulation of equation (3) yields a new reliability expression, $R_{upper}(V)$,

$$R_{upper}(V) = P(R_{old}(V)) + P(S_1, I_1)$$

$$- P(R_{old}(V)) P(R_{old}(V))$$

This new recursive equation yields a high estimate for the exact vertex reliability $R(V)$. To illustrate this, let $S_i, l = 1, \ldots, n$, denote the event that source node $i$ of vertex $V$ is connected to the root node by at least one simple path. Similarly, let $I_i, i = 1, \ldots, n$, represent the event that link $i$ is operational. It can be shown that the probability of finding one or more (arbitrarily defined) sets of links that are operational, given that another set is known beforehand to be operational, is greater than or equal to the probability of finding such sets given no a-priori knowledge of the status of any links [2]. In other words, knowing that a given group of links is operational increases the probability that a randomly selected link will be operational, since it may be a member of this group. From this relationship, it is then proven that if at least one node is known beforehand to be joined to the root node by at least one simple path, then the probability increases that another randomly selected node is similarly joined to the same root node. These ideas, which serve as the genesis of theorem 3 of [2], result in the following inequality,

$$P(S_1, I_1 \cup \ldots \cup S_n, I_n) \geq P(S_1, I_n)$$

The substitution of equation (2) into (6) yields the inequality

$$P(S_1, I_1 | R_{old}(V)) \geq P(S_1, I_1)$$

This expression shows that the replacement of the conditional probability $P(S_1, I_1 | R_{old}(V))$ by $P(S_1, I_1)$ results in a low estimate for the joint probability $P(S_1, I_1, R_{old}(V))$. Substituting (7) into the exact expression of (3) and comparing the result with (5) gives the desired result,

$$R_{upper}(V) \geq R(V)$$

A Lower Bound

An approach similar to that used in the preceding section yields a lower bound for
R(V). Consider the conditional probability
\[ P(S_1 | R_{old}(V)) \]
of equation (3). If an upper
bound for this conditional probability, say
\[ P_{max}(S_1 | R_{old}(V)) \]
could be found, then the
substitution of such a bound into the exact
reliability expression of equation (3) would
generate a lower bound, \( R_{lower}(V) \), for the
source-to-terminal reliability \( R(V) \),
\[ R_{lower}(V) = P(R_{old}(V)) + P(S_1) - P_{max}(S_1 | R_{old}(V)) P(R_{old}(V)) \]
where
\[ R_{lower}(V) \leq R(V) \]
The above inequality is, of course,
predicated on the assumption that an
appropriate upper bound, \( P_{max}(S_1 | R_{old}(V)) \),
is available. From probability theory,
\[ P(S_1 | R_{old}(V)) = 1 - P(S_1 | R_{old}(V)) \]
which implies that finding an upper bound
for \( P(S_1 | R_{old}(V)) \) is tantamount to finding
a lower bound for \( P(S_1 | R_{old}(V)) \). Using the
definition of \( R_{old}(V) \) in equation (2), this
latter quantity is
\[ P(S_1 | R_{old}(V)) = P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) \]
which is the desired lower bound for
\[ P(S_1 | R_{old}(V)). \]

The quantity \( P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) \)
in equation (13) represents the probability
that the event-pair \( S_1 \) does not occur,
given that all event-pairs which define
\( R_{old}(V) \) have already occurred. If all such
event-pairs \( S_1, i = 1, \ldots, n-1 \) have indeed
occurred, then all these source nodes \( S_i \) are
connected to the root node by at least one
simple path and the corresponding \( i \) links,
incident into vertex \( V_i \), are also all
operational. Under these conditions, the
failure of the event-pair \( S_1 \) can occur due
to a variety of reasons. One approach to
guarantee that the event-pair \( S_1 \) does not
occur, given that the compound event
\( R_{old}(V) \) occurs, is for link \( i \) to fail.

Another way to ensure the failure of this
event, given the occurrence of \( R_{old}(V) \), is to
have all incident links into source node \( n \) fail
and have all links emanating from this
source node that are incident into other
source nodes of vertex \( V_{(n)} \) contained in the
compound event \( R_{old}(V) \), fail. This is
necessary to ensure that the individual
event-pairs which comprise \( R_{old}(V) \) do not
occur via a path through source node \( n \),
thereby allowing this node to be connected
to the root node. The failure modes that
are allowed by the conditional probability
of equation (13) are those that allow event
\( R_{old}(V) \) to occur, but prohibit the
simultaneous occurrence of the event-pair
\( S_1 \). Since there are additional causes for
the failure of event-pair \( S_1 \) to occur, given
that all event-pairs which define
\( R_{old}(V) \) have already occurred, \( P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) \) can be expressed in terms of two
mutually exclusive events as
\[ P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) = P(S \text{ fails as described} U 1, \text{ fails}) + P(S_1 \text{ fails due to other reasons}) \]
which suggests the inequality
\[ P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) \geq P(S_1 \text{ fails as described} U 1, \text{ fails}) \]
The quantity \( P(S_1 \text{ fails as described} U 1, \text{ fails}) \) in (15) above is the new lower bound
for \( P(S_1 \in (S_{in} \cup U \ldots U S_{n-1,n})) \) and can be evaluated in a
straightforward manner. The probability
that source node \( n \) is completely
disconnected is the probability that all
incident links into it fail and the
probability that all links from it to other
source nodes of vertex \( V_{(n)} \), which are
contained in the compound event \( R_{old}(V) \),
simultaneously fail. Under the assumption
that all links fail independently with equal
probability \( q = 1-P(l) \), this quantity can
be written as
\[ P(S_1 \text{ fails as described}) = q^{\text{indeg}(S_1)} q^{R(S(V,S_1))} \]
where \( \text{indeg}(S_1) \) is the indegree of node \( S_1 \)
and \( R(S(V,S_1)) \) is the number of links
emanating from \( S_1 \) incident into source nodes
referred to the compound event \( R_{old}(V) \).
Since the probability that link 1 fails is also given by \( q \) and all link failures are statistically independent, the probability of the union of these two events, \( P(S_1 \cap U \text{ link fails}) \), is given by

\[
P(S_1 \cap U \text{ link fails}) = q + q^\text{indeg(S)\(N\)+1}=R(V,\text{in})
\]

\[
- q^\text{indeg(S)\(N\)+1}=R(V,\text{in})
\]

Equation (17) above establishes the existence of a lower bound on \( P(S_1 \text{ fails as described \(U \text{ link fails})} \)

\[
\text{for } \sum_{i=1}^{N} R(S_i \text{ fails as described \(U \text{ link fails})} \)
\]

A sufficient condition for the existence of a lower bound on \( R(V) \).

**NUMERICAL RESULTS**

**Time Complexity**

The time complexity of the bounding algorithm is bounded by a linear function of the product of the number of network nodes and the square of the maximum indegree of the network nodes,

\[
\text{Execution Time} \leq C \cdot N \cdot \text{indeg}_{\text{max}}^2
\]

where \( N \) is the number of network nodes, \( \text{indeg}_{\text{max}} \) is the maximum indegree of the network nodes and \( C \) is a constant. Fixing \( \text{indeg}_{\text{max}} \), renders the execution time a linear function of the number of network nodes, \( N \). Typical values for \( \text{indeg}_{\text{max}} \) are 16 and 32 for concentrators in an Ethernet-type environment, so this restriction is not unreasonable. In any event, the maximum possible indegree of any node in an arbitrary network is \( (N-1) \) for a fully-connected \( N \) node network, which renders the execution time a cubic function of \( N \). Although not as rapid as a linear-time algorithm, a cubic-time method is sufficiently tractable and is a significant and welcome improvement over previous exponential-time schemes.

The time-complexity of the bounding algorithm was empirically explored by determining the source-to-terminal reliabilities of various digraphs of increasing size. The networks for these test cases were randomly generated and ranged in size from 5 nodes and 10 links to 20 nodes and 50 links. The results for the various digraphs are given in table 1 below.

**TABLE 1: SOLUTION TIME FOR SAMPLE DIGRAPHS**

<table>
<thead>
<tr>
<th>Graph</th>
<th>Nodes</th>
<th>Links</th>
<th>Solution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>20</td>
<td>50</td>
<td>10.82 sec</td>
</tr>
<tr>
<td>#2</td>
<td>16</td>
<td>40</td>
<td>9.32 sec</td>
</tr>
<tr>
<td>#3</td>
<td>12</td>
<td>30</td>
<td>6.73 sec</td>
</tr>
<tr>
<td>#4</td>
<td>8</td>
<td>20</td>
<td>4.72 sec</td>
</tr>
<tr>
<td>#5</td>
<td>5</td>
<td>10</td>
<td>2.09 sec</td>
</tr>
</tbody>
</table>

The results clearly demonstrate that the time required to obtain an estimate for the source-to-terminal reliability is a linear function of the network size. This relationship identifies the number of nodes as the important metric of network complexity, while the number of links indirectly plays a role in terms of the a-priori limitations \( \text{indeg}_{\text{max}} \).

**Demonstration of Algorithm Accuracy**

The linear-time complexity of the bounding algorithms alone is insufficient justification for its use over previous techniques. Clearly, if the bounding methods are not sufficiently accurate then the computational advantages of such schemes are totally negated by the knowledge that the answers generated will be nearly useless.

The experimental verification of the accuracy of any calculation procedure requires its application to a variety of sample problems, whose exact solutions are known beforehand. Unfortunately, the nature of intractable problems precludes the inclusion of excessively large cases in such a collection of sample problems, since the exact solution cannot be computed within a reasonable length of time. These tractability difficulties placed a severe upper bound on the number of network nodes that could be allowed in any sample network. Other networks of moderate size have been analyzed in the literature \([1,3,9,10]\), but they have for the most part been undirected graphs and the results obtained were for overall reliability rather than for s-t reliability. Networks from such sources were selected as being topologically representative and were rendered directed by the addition of arrowheads where appropriate. Two graphs from Soi and Aggarwal \([11]\) and one from Bailey and Kulkarni \([1]\) were analyzed in terms of the s-t reliability for the lowest-numbered node pairs. The sample graphs are given in figures 1 through 3 and the corresponding upper and lower bounds, along with the exact results, are given in tables 2 through 4, respectively.

The computation of both upper and lower bounds for s-t reliability also implicitly generates an approximate answer. If the upper and lower bounds are averaged, then the result can serve as an estimate of the actual s-t reliability. This parameter was calculated along with the bounds, and the percentage deviation of this figure from the exact s-t reliability was computed and is given in the last column of tables 2 through 5. While no claims are made as to the biasedness and other statistical properties of this estimate, it does serve as a rough indication of the relative "tightness" of the absolute bounds and may have some merit.
The most important observation regarding the data is that the exact answer, given in the third column, does indeed lie between the lower and upper bounds computed by the algorithm. Although hardly unexpected, the empirical confirmations that these numerical results supply are a necessary step to foster trust in the reliability limits generated by the approximation procedure.

Another important observation is that the percent deviations are rather small, indicating that the bounds generated are relatively "tight". As a comparison, Soi and Aggarwal [11] present an approximate method for overall network reliability for which no computation time claim is made. The accuracy of their method is on the order of a few percent for values of link reliability of 0.9, and is heavily dependent upon the clustering scheme chosen for a given network topology. Although their experimental results show that the error decreases asymptotically to 0 for link reliabilities of 1.0, the same is true of both the upper bound (for link reliabilities of 1.0) and of the lower bound (for link reliabilities of 0.0). In addition, no absolute guarantees are made as to the maximum possible estimation error in [11] whereas such a promise is automatically available by the linear-time bounding method presented here.

Although the preceding tables only
present the s-t reliability for one source-termina pair, the s-t reliabilities for all N s-t pairs, as seen from the root node, are simultaneously generated. Thus, to obtain bounds on the s-t reliabilities for all possible s-t node pairs, the bounding algorithm need only be invoked N times, for all possible root nodes. With the maximum indegree of all nodes fixed, the number of computations involved in this procedure is O(N), which is a considerable improvement over exponential schemes.

CONCLUSIONS

An algorithm to compute the upper and lower bounds for the source-to-terminal reliability of all nodes in a computer network is presented. These source-to-terminal reliabilities are defined with respect to an arbitrary starting (or root) node and represent the probabilities of reaching each of the network nodes from the root node. A unique feature of this algorithm is that it possesses linear time-complexity when the maximum indegree of all network nodes is limited.

REFERENCES


