Exact calculation of computer network reliability

by E. HÄNSLER
IBM Research Laboratory
Ruschlikon, Switzerland

G. K. McAULIFFE
IBM Corporation
Dublin, Ireland

and

R. S. WILKOV
IBM Corporation
Armonk, New York

INTRODUCTION

The exact calculation of the reliability of the communication paths between any pair of nodes in a distributed computer network has not been feasible for large networks. Consequently, many reliability criteria have been suggested based on approximate calculations of network reliability. For a thorough treatment of these criteria, the reader is referred to the book and survey paper by Frank and Frisch 1,2 and the recent survey paper by Wilkov. 3

Making use of the analogy between distributed computer networks and linear graphs, it is noted that a network is said to be connected if there is at least one path between every pair of nodes. A (minimal) set of links in a network whose failure disconnects it is called a (prime) link cutset and a (minimal) set of nodes with the same property is called a (prime) node cutset. If a node has failed, it is assumed that all of the links incident to that node have also failed. A cutset with respect to a specific pair of nodes ns and nt in a connected network, sometimes called an s-t cut, is such that its removal destroys all paths between nodes ns and nt.

The exact calculation of \( P_e[s, t] \), the probability of successful communication between any pair of operative computer centers \( n_s \) and \( n_t \), requires the examination of all paths in the network between nodes \( n_s \) and \( n_t \). More specifically, if each of the \( n \) nodes in any given network fail with the same probability \( q \) and each of the \( b \) links fail with the same probability \( p \), then \( P_e[s, t] \) is approximately given by

\[
P_e[s, t] = \sum_{i=0}^{b} A^*_{ns}(i) (1-p)^i p^{b-i}, \quad p \gg q.
\]

In Eq. (1), \( A^*_{ns}(i) \) is the number of combinations of \( i \) links such that if only they are operative, there is at least one communication path between nodes \( n_s \) and \( n_t \). On the other hand, the calculation of the probability \( P_f[s, t] \) of a communication failure between nodes \( n_s \) and \( n_t \) requires the examination of all s-t cuts. For specified values of \( p \) or \( q \), \( P_f[s, t] \) is approximately given by

\[
P_f[s, t] = \sum_{i=0}^{b} C^*_{ns}(i) p^i(1-p)^{b-i}, \quad p \gg q.
\]

For \( q \gg p \), a similar expression can be given replacing \( C^*_{ns}(i) \) by \( C^*_{nt}(i) \). The coefficients \( C^*_{ns}(i) \) and \( C^*_{nt}(i) \) denote the total number of link and node s-t cuts of size \( i \). The enumeration of all paths or cutsets between any pair of nodes \( n_s \) and \( n_t \) is not computationally possible for very large networks.

RELIABILITY APPROXIMATION BASED ON CUTSET ENUMERATION

If any network \( G \) of \( b \) links and \( n \) nodes, it is easily shown that the order of the number of cutsets is \( 2^{n-1} \)
From the collection of the Computer History Museum (www.computerhistory.org)

whereas the order of the number of paths between any pair of nodes is $2^{2-n+i}$. For networks having nodes of average degree (number of incident links) greater than four, $b>2n$ and $2^{2-n+i}>2^{-r}$. Consequently, such networks have a larger number of paths than cutsets. Computation time would clearly be reduced in such cases by calculating network reliability from cutsets instead of paths. In this case $P^*[s, t]$ can be obtained from $P^*[s, t] = 1 - P^*[s, t]$, where $P^*[s, t]$ can be calculated from Eq. (2). Alternatively,

$$P^*[s, t] = P^*[\bigcup_{i=1}^N C^i_{s,t}]$$

where $C^i_{s,t}$ is the event that all links fail in the $i$th prime $s$-$t$ cut and $N$ is the total number of prime cutsets with respect to nodes $n_s$ and $n_t$. The calculation of $P^*[s, t]$ from Eq. (2) clearly requires the examination of all $s$-$t$ cuts. The number of prime $s$-$t$ cuts is usually much smaller. However, $P^*[s, t]$ is not readily calculated from Eq. (3) because the $C^i_{s,t}$ are not mutually exclusive events.

Following Wilkov, we shall use $P^*_s = \text{Max}_s, P^*[s, t]$ as an indication of the overall probability of service disruption for a given computer network. For specified values of $p$ or $q$, $P^*_s$ depends only on the topology of the network. A maximally reliable network clearly has a topology which minimizes $P^*_s$ and hence minimizes $\text{Max}_s, C^s_{s,t}(m)$ or $\text{Max}_s, C^s_{s,t}(m)$ for small (large) values of $m$ when $p$ or $q$ is small (large). Letting $X^s_{s,t}(m)$ and $X^s_{s,t}(m)$ denote the number of prime node and edge $s$-$t$ cuts of size $m$, $X^s(m) = \text{Max}_s, X^s_{s,t}(m)$ and $X^s(m) = \text{Max}_s, X^s_{s,t}(m)$ have been proposed as computer network reliability measures. These measures $X^s(m)$ and $X^s(m)$ denote the maximum number of prime node and edge cutsets of size $m$ with respect to any pair of nodes. A maximally reliable network is such that $X^s(m)$ and $X^s(m)$ are as small as possible for small (large) values of $m$ when the probability of node or link failure is small (large).

In the calculation of $X^s(m)$ and $X^s(m)$ for any given network, all node pairs need not be considered if all nodes or links have the same probability of failure. It has been shown that in order to calculate $X^s(m)$ and $X^s(m)$, one need only consider those node pairs whose distance (number of links on a shortest route between them) is as large as possible. For a specified pair of nodes $n_s, n_t, X^s_{s,t}(m)$ can be calculated for all values of $m$ using a procedure given by Jensen and Bellmore. Their procedure enumerates all prime link cutsets between any specified pair of nodes in a non-oriented network (one consisting only of full or half duplex links). It requires the storage of a large binary tree with one terminal node for each prime cutset. Although these cutsets are not mutually exclusive events, it has been suggested that Eq. (3) be approximated by

$$P^*[s, t] = \sum_{i=0}^N P^*[C^i_{s,t}]$$

However, it is shown in the following section that no additional computation time is required to actually compute $P^*[s, t]$ exactly.

**EXACT CALCULATION OF COMPUTER NETWORK RELIABILITY**

A simple procedure is described below to iteratively calculate a minimal set of mutually exclusive events containing all prime link $s$-$t$ cuts. This procedure starts with the prime cutset consisting of the link incident to node $n_t$. Subsequently, these links are re-connected in all combinations and we then cut the minimal set of links adjacent to these that lie on a path between node $n_t$ and $n_t$, assuming that the network contains no pendant nodes (nodes with only one incident link). The link replacements are iterated until the set of links connected to node $n_t$ are reached. The procedure is easily extended to provide for node cutsets as well and requires a very small amount of storage since each event is generated from the previous one. $P^*[s, t]$ is obtained by accumulating the probabilities of each of the mutually exclusive events.

**Procedure I**

1. Initialization

   Let: $N$ be the set of all nodes except nodes $n_t$.
   \[ C = \text{the set of all links not incident to node } n_t. \]
   \[ M_1 = [n_t] \]
   \[ F_1 = \text{the set of links incident to both } n_t \text{ and } n_t. \]
   \[ S_1 = \text{the set of links incident to } n_t \text{ but not } n_t. \]
   \[ b_1 = \text{a binary number consisting of only } |S_1| \text{ ones}. \]
   \[ i = 1 \]

2. Let:

   \[ T_i = \text{be a subset of } S_i \text{ consisting of those elements in } S_i \text{ for which the corresponding digit in } b_i \text{ is unity}. \]
   \[ M_{i+1} = \text{be a subset of } N \text{ consisting of nodes incident to the links in } T_i. \]
   \[ N = N - M_{i+1}. \]
   \[ F_{i+1} = \text{be a subset of } C \text{ consisting of links incident to } n_t \text{ and adjacent to any member of } T_i. \]
   \[ S_{i+1} = \text{be a subset of } C \text{ consisting of links incident to nodes in } N \text{ other than } n_t \text{ and adjacent to any member of } T_i. \]
   \[ C = C - (S_{i+1} \cup F_{i+1}). \]
3. If $S_{i+1} \neq \emptyset$, then let:
   
   \begin{align*}
   b_{i+1} & \quad \text{be a binary number with } |S_{i+1}| \text{ ones} \\
   i & = i+1
   \end{align*}
   
   Go to step 2
   
   Otherwise, let:
   
   \begin{align*}
   T_{i+1} & = \emptyset \\
   CS & = \cup \left[ F_k \cup \hat{T}_k \cup (S_k - T_k) \right], \\
   C & = CUF_{i+1} \cup S_{i+1}
   \end{align*}
   
   where $CS$ is a modified cutset and $\hat{T}_k$ indicates that the links in set $T_k$ are connected.
   
   4. Let:
   
   \begin{align*}
   C & = C \cup F_{i+1} \cup S_{i+1} \\
   N & = N \cup M_{i+1} \\
   b_i & = b_{i-1} \text{ (modulo 2)} \\
   & \text{If } b_i < 0, \text{ go to step 5. Otherwise, go back to step 2.}
   \end{align*}
   
   5. Let $i = i-1$. If $i \neq 0$, go back to step 4. Otherwise, terminate the procedure.

In the calculation of $P_{\gamma}[s,t]$, Procedure I performs a depth first search of the given network starting at node $n_s$ and traversing several links at the same time. The index $i$ indicates how far from $n_s$ the search has progressed and $b_i$ indicates the links traversed at the $i$th level of the search. During the search, set $N$ keeps track of the nodes which have not yet been reached and $C$ is the set of links not yet traversed. At the $i$th level, set $F_{i+1}$ is a subset of the links not yet traversed which are incident to node $n_i$, and hence must be disconnected in the formation of an $s-t$ cut. Set $S_{i+1}$ consists of edges in $C$ which lie on a path to $n_i$ but which need not necessarily be disconnected in the formation of an $s-t$ cut. The links in $T_{i+1} \subseteq S_{i+1}$ are those which are connected as we traverse the network toward node $n_i$. When set $S_{i+1}$ is empty, the edges incident to $n_i$ have been reached, and this portion of the search is terminated with the formation of a modified $s-t$ cut in step 3 of the procedure. The modified $s-t$ cut is actually a group of states in the network or an event in which all links in an $s-t$ cut are disconnected and the links in all $T_i$ in this part of the search are connected. It is the set of connected links which makes this modified $s-t$ cut mutually exclusive of all of the modified $s-t$ cuts previously generated during the execution of Procedure I. In step 4, we back track one level and then continue the search by traversing a different subset of the links in $S_i$. After all combinations of the links in $S_i$ have been traversed, we back track one additional level and the search continues with traversal of a different combination of the links in $S_{i+1}$. The procedure terminates when we have back tracked all the way up to node $n_s$. It is shown in the proof of the following theorem that the modified $s-t$ cuts generated are mutually exclusive and collectively exhaustive.

**Theorem I:**

Procedure I generates a collectively exhaustive set of mutually exclusive modified $s-t$ cuts.

**Proof:**

**Part I—Prime $s-t$ cuts**

In this part of the proof, it is shown that every modified cutset $CS$ generated in step 3 of Procedure I contains a prime $s-t$ cut. We begin by noting that the links in $T_k (1 \leq k \leq i+1)$, traversed in the depth first search through the given network, form subnetworks containing node $n_s$. For any such subnetwork, set $M_i$ contains the nodes at a distance of $i$ from $n_s$. Each modified cutset $CS$ generated by the procedure consists of all links in such a subnetwork being connected and all links in $S_i - T_k$ and $F_s$ that connect nodes inside the subnetwork with those outside the subnetwork, being disconnected. Node $n_s$ is never contained in the subnetwork since any link incident to $n_s$ must be contained in some set $F_s$ and is therefore always disconnected.

**Part II—Mutually exclusive**

In order to show that the modified cutsets obtained from Procedure I are mutually exclusive, we shall demonstrate that every pair of modified cutsets disagree in the state of at least one link appended at some level $j$. Specifically, for any pair of distinct modified cutsets $CS_p$ and $CS_q$ there exists a value of $j$ for which if $\hat{T}_j \subset CS_p$ and $\hat{T}_j \subset CS_q$, then $T_k = T_k'$ for $k \leq j - 1$ but $T_j \neq T_j'$. This implies that during the generation of $CS_p$ and $CS_q$, from Procedure I, $b_k (k \leq j - 1)$ was the same in both cases but the value of $b_j$ was different. Otherwise, if $CS_p$ and $CS_q$ were generated from the same values of $b_k$ for all $k$, then $CS_p$ and $CS_q$ would be identical. It is now noted that if $T_j \neq T'_j$, there exists a link $e$ such that $e \in T_j$ and $e \notin T'_j$ which implies that $e \notin (S_j - T'_j)$. It follows that link $e$ appears connected in $CS_p$ and disconnected in $CS_q$.

**Part III—Collectively exhaustive**

We shall prove that the modified $s-t$ cuts obtained from Procedure I are collectively exhaustive by showing that every state of the network in which nodes $n_s$ and $n_t$ cannot communicate is contained in one of the modified $s-t$ cuts. We proceed by noting that in any given state of the network that includes an $s-t$ cut, there is a maximal set of nodes $N_i$ connected to node $n_s$ which does not include node $n_t$. If set $N$ were discarded in Procedure I, then the resulting modified $s-t$ cuts would contain every state of the links on set $N_i$ in which there is a path between every pair of nodes in...
This follows from the fact that in the generation of all modifications of the same prime s-t cut, the deletion of set \( N \) from Procedure I would result in set \( S_k \) for all \( k \) containing every link on set \( N \), As we sequence through all \( b_n \), these links would be connected and disconnected in Procedure I in every combination in the traversal of all paths from \( n_s \) to every other node in \( N \). It is now noted that any modified s-t cut generated from Procedure I that includes a connected subnetwork on \( N \), specifies as cut all links connecting nodes \( N_s \) to nodes in \( N-N_s \), where \( N \) is the set of all nodes in the network. All links in the network connecting pairs of nodes in \( N-N_s \) would be unspecified.

Taking advantage of the unspecified links, it is possible to extend one of the modified s-t cuts generated by Procedure I with set \( N \) deleted to match any specified state of the network in which nodes \( n_s \) and \( n_t \) are not connected. The effect of setting \( N \) in Procedure I is to omit several links from many of the \( S_k \). Significantly fewer modified s-t cuts are thereby generated since the states of the redundant links joining pairs of nodes in \( N \) would not be specified. However, these modified s-t cuts clearly include all of those generated when set \( N \) is neglected. This is evident since each of the links on \( N \), not specified can be assigned a particular state in order to match a given modified s-t cut obtained from Procedure I with set \( N \) omitted. Consequently, any specified state of the network containing an s-t cut is included in one of the modified s-t cuts obtained from Procedure I. Q.E.D.

It should be noted that the collectively exhaustive set of mutually exclusive modified s-t cuts obtained from Procedure I is not minimum. This is due to the fact that for any prime s-t cut, Procedure I as given generates too many subnetworks on the set of nodes \( N \), connected to \( n_s \). However, Procedure I is easily modified to eliminate the generation of any subnetworks on \( N \), that contain circuits. This is done by eliminating all \( T_i \) in step 2 of the procedure in which two or more links are incident to the same node. The formation of any other circuits in subnetworks on \( N \), is avoided through the use of set \( N \) in Procedure I. The result is that the connected links in any modified s-t cut would form trees on \( N \).

It is noted that in the procedure given above, nodes have been assumed to be perfectly reliable. However, Procedure I can also be applied in the case that nodes fail and links are perfectly reliable. In the event that nodes and links may fail simultaneously, assuming that their failures are statistically independent, following H"{a}nsler we can easily modify Procedure I to obtain a collectively exhaustive set of mutually exclusive modified s-t cuts consisting of nodes and links. We would proceed by introducing a binary number \( b^* \) consisting of only \( M_i \) ones for each of the sets \( M_i \) in Procedure I. Analogous to \( T_i \), in step 2 we form a set \( T_{i+1}^* \) consisting of the nodes in \( M_{i+1} \) for which the corresponding digit in \( b^* \) is unity. \( F_{i+1} \) and \( S_{i+1} \) in step 2 would consist of links in \( C \) incident to nodes in \( T_{i+1}^* \). Then any modified s-t cut \( CS \) formed in step 3 of Procedure I would consist of

\[
CS = \bigcup_{i=1}^{i+1} \left[ F_i \cup T_i \cup (S_i - T_i) \right] \bigcup_{i=1}^{i+1} \left[ F_i \cup T_i \cup (S_i - T_i) \right]
\]

The only other s-t cut consists of node \( n_t \) being inoperative. The above modifications to Procedure I double the number of levels and therefore significantly increase the necessary computation time for any given network. However, the storage requirement of the modified procedure is still very small. A network of \( b \) links and \( n \) nodes would only require approximately \( 3b+2n \) words of storage to compute \( P[s,t] \) in the presence of node and link failures. All modified cutsets are either printed out or their probabilities accumulated. Consequently, the exact calculation of \( P[s,t] \) for any given network is limited only by the computer time required in view of the inherent computational complexity of the problem.

**EXAMPLES OF NETWORK RELIABILITY CALCULATIONS**

In this section, Procedure I will be used to obtain \( P[s,t] \) for several networks, assuming that all nodes are perfectly reliable and all links fail with the same probability \( p \). We shall first consider the simple network shown in Figure 1 in order to demonstrate the modified 1-4 cuts obtained from Procedure I. Figure 1

![Figure 1](image)

**Prime 1-4 Cuts**

<table>
<thead>
<tr>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>

**Modified 1-4 Cuts**

<table>
<thead>
<tr>
<th>5</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
<tr>
<td>a</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>

Figure 1—Example illustrating the calculation of node pair failure probability.
Figure 2—Example for comparison of approximate and exact reliability calculations

shows the four prime cutsets between nodes 1 and 4, which are not mutually exclusive. Also listed there are the six mutually exclusive modified 1-4 cuts obtained from Procedure I in the order in which they are obtained. Note that the second and fourth modified 1-4 cuts are not prime since link a or b has been disconnected in order for the corresponding events to be mutually exclusive.

The network shown in Figure 2 has been given by Jensen and Bellmore as an example of their procedure for enumerating all prime cutsets with respect to a given pair of nodes. They listed 16 prime 1-8 cuts for the network of Figure 2. From these cutsets, \( P_r[1, 8] \) was approximated by

\[
P_r[1, 8] \approx 4p^3 + 8p^3 + 4p^4
\]

(6)

However, from the mutually exclusive modified 1-8 cuts obtained from Procedure I, \( P_r[1, 8] \) is actually given by

\[
P_r[1, 8] = 4p^3 + 6p^4 - 16p^5 - 32p^6 + 115p^7 - 134p^8 + 79p^9 + 24p^{10}
\]

(7)

It is clear from this example that the approximation to \( P_r[8, 1] \) given by Jensen and Bellmore is reasonable

| TABLE I—Polynomial Coefficients for \( P_r[9, 6] \) for Networks of Figure 3 |
|-----------------|-----------------|-----------------|
| Coefficient    | \( 3 \)          | \( 2 \)          |
| \( c_4 \)       | \( 6 \)          | \( 4 \)          |
| \( c_5 \)       | \( -6 \)         | \( 9 \)          |
| \( c_7 \)       | \( -25 \)        | \( -22 \)        |
| \( c_7 \)       | \( -25 \)        | \( -153 \)       |
| \( c_9 \)       | \( 257 \)        | \( 572 \)        |
| \( c_9 \)       | \( -417 \)       | \( -874 \)       |
| \( c_{10} \)    | \( 364 \)        | \( 744 \)        |
| \( c_{10} \)    | \( -427 \)       | \( -829 \)       |
| \( c_{11} \)    | \( 46 \)         | \( 102 \)        |
| \( c_{12} \)    | \( -5 \)         | \( -12 \)        |

Figure 3—ARPA subnetwork topologies having 9 nodes and 12 links: (a) actual, (b) example based on \( X^r(m) \)

Figure 4—ARPA subnetwork topologies having 15 nodes and 19 links: (a) actual, (b) example based on \( X^r(m) \)
for only very small values of $p$ since only the first coefficient in that approximation is exact.

Two topologies for 9 node and 15 link subnetworks of the ARPA network are shown in Figure 3. The network shown in Figure 3a was given by Frank, et al. Figure 3b is a maximally reliable network based on $X_n(m)$ and $X_e(m)$ for small $m$ obtained by Wilkov. Assuming all nodes are perfectly reliable and all links fail with the same probability $p$, $P_f[9, 6]$ can be expressed as

$$P_f[9, 6]= \sum_{i=2}^{12} c_i p^i. \quad (8)$$

The coefficients in Eq. 8 for Figures 3a and 3b are listed in Table I. They have been obtained in 18 seconds using an APL implementation of Procedure I on a 360 model 91 computer. Consistent with the results in Reference 9, Figure 3b has smaller coefficients than Figure 3a for small powers of $p$. Furthermore, we have found that there are a total of 2,772 cutting states with respect to nodes 9 and 6 in Figure 3b compared with 3,011 in Figure 3a. Similar results have been obtained for the 15 node and 19 link ARPA subnetwork topologies shown in Figure 4. The topology shown in Figure 4a was given by Frank, et al. and Figure 4b was obtained by Wilkov based on $X_n(m)$ and $X_e(m)$. The polynomial coefficients for $P_f[15, 2]$ are given in Table II. The total number of cutting states between nodes 15 and 2 is 49.7 thousand for Figure 4a and 44.9 thousand for Figure 4b.

CONCLUSION

A procedure has been given for calculating the node pair failure probability in computer networks exactly, using little more computation time than previously required to obtain an upper bound on $P_f[s, t]$. Furthermore, the storage requirement of the given procedure grows only linearly with the number of links in the given network. Unfortunately, due to the inherent computational complexity of the problem, the necessary computation time grows exponentially with the size of the given network. Nonetheless, it has been found to be computationally feasible to use the procedure given herein for networks as large as the ARPA network.

REFERENCES

1. H Frank I T Frisch
   *Communication, transmission, and transportation networks*
   Addison-Wesley Publishing Company Reading Massachusetts 1971

2. H Frank I T Frisch
   *Analysis and design of survivable networks*

3. R S Wilkov
   *Analysis and design of reliable computer networks*
   IEEE Transaction on Communications Vol COM-20 June 1972 pp 660-628

4. R S Wilkov
   *Reliability considerations in computer network design*
   Proceedings of IFIP Congress '71 Ljubljana Yugoslavia August 1971

5. R S Wilkov
   *On the design of maximally reliable communication networks*
   Proceedings of the Sixth Annual Princeton Conference on Information Sciences and Systems March 1972

6. P A Jensen M Bellmore
   *An algorithm to determine the reliability of a complex system*

7. E Hansler
   *A fast recursive algorithm to calculate the reliability of a communication network*

8. H Frank et al
   *Store and forward computer networks*

9. R S Wilkov
   *Design of computer networks based on a new reliability measure*