BDD Method for Evaluating Reliability of Traffic-Path-Based Network Model

Takuto Koizumi, Masahiro Hayashi †

Department of Electrical, Electronics and Comunication Engineering, TOKYO CITY UNIVERSITY
Tokyo, Japan
† E-mail: mhaya@tcu.ac.jp

Abstract—**This paper proposes a new method for evaluating the reliability of traffic-path-based models representing telecommunications networks. The key idea is to use the binary decision diagram (BDD) method, which memorizes the computational results in the binary expansion appearing in the evaluation, so that we can reuse them in further steps. While this method is a fast evaluation for other models such as graph-based models, no previous work has applied it to a traffic-path-based network model. Our numerical experimental results show that the BDD method is surely faster for large models than the most recently proposed existing method.**

Key Words—Reliability, Telecommunications networks, Traffic-path-based network model, BDD, Algorithm

I. Introduction

Studies for how to evaluate reliabilities of telecommunications networks are categorized in two types. One type focuses on 'graph-based network model', where graph is mathematical concept consisting of nodes and links [1]-[3]. The other is 'traffic-path-based network model', where it does not use graph but use traffic-path expressing transmission or logical paths actually used in actual telecommunications networks.

Refs. [4][5][6] emphasized that traffic-path-based network model is very useful in reliability design process, because it can express the actual mechanisms that cause failures in telecommunications networks.

Nevertheless, this model still faces a problem because the execution times of the reliability evaluation algorithms [4][6] for this model increase exponentially with the number of traffic paths (i.e., the evaluation is an NP-hard problem) or only approximation [5].

Now, this paper proposes a new approach which is more efficient than the previous fastest algorithm [6] for an exact evaluation. In our proposal, the traffic-path-based network model incorporates the binary decision diagram (BDD) method. While this method has been used in the reliability engineering field [2][7], it never has been used in traffic-path-based network modelling.

II. Previous Work

A. Preparation

The special symbols and terms used throughout this paper are defined below.

Pr( ): The probability of the occurrence of an event in ( )

ϕ: Empty set

$\rho(S)$: The power set of S for any finite set S, where $\rho(S)$ is the set of all subsets of S ($\phi \in \rho(S)$)

$|S|$: The number of elements of S

R^+: The set of non-negative real numbers

$S_1 - S_2$: The elements included in S_1 but not included in S_2
For example, $\{1, 2, 3\} - \{2, 3, 4\} = 1$.

'iff' is used to mean 'if and only if'.

B. Definition of model

Let model N be $N = (X, Y \varphi, C_0)$, where X, Y, φ, C_0 are defined as follows.

X: A finite set
(This represents the set of physical elements (routers, cables, and other equipment) of a telecommunication network.)

Y: A finite set
(This represents the set of traffic paths, where a traffic path is sequence of physical elements through which to convey the traffic of a telecommunication network.)

φ: A mapping from Y to $\rho(X)$
(For $y \in Y$, $\varphi(y)$ is the set of physical elements that traffic path y goes through.)

C_0: A mapping from $\rho(Y)$ to R^+
(For $A = \{y_1, y_2, \ldots, y_n\} \in \rho(Y)$, $C_0(A)$ represents the capacity of traffic ensured when all traffic paths in A work.)

Let us describe an example of model N, representing the case of Fig. 1. (In the cases of Figs. 1- 4, we assume that only cables which are illustrated by lines fail. This is assumed for simplicity. We emphasize that model N can be applied to cases in which all physical elements can fail.)

Example network:

$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, $Y = \{y_1, y_2, y_3\}$

$\varphi(y_1) = \{x_1, x_6\}$, $\psi\{y_2\} = \{x_1, x_5, x_7, x_8\}$, $\varphi\{y_3\} = \{x_3, x_4, x_7, x_8\}$

In this example, the value of $C_0(\ )$ is computed under the following assumptions.

Assumption I. The capacity of each traffic path is 100.

© Copyright IEICE – The 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) 2019

Assumption II. y_1 & y_2 are used as primary traffic paths. Traffic-path y_3 is used as a back-up.

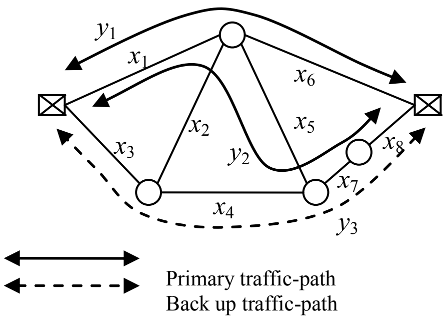

Fig. 1 Example network.

From these assumptions, C_0 is determined as follows.

1. If $A = \phi$, then $C_0(A) = 0$,
2. else if $A = \{y_1\}$ or $A = \{y_2\}$ or $A = \{y_3\}$, then $C_0(A) = 100$;
3. else $C_0(A) = 200$.

Furthermore, we make the assumptions below for model N.

Assumption 1. For any $x_i \in X$, x_i has two states: up (working) and down (failed).

Assumption 2. The probability of any $x_i \in X$ being down is q_i. The probability of x_i being up $p_i \equiv 1 - q_i$.

Assumption 3. For any $x_i \in X$, and any $x_j \in X$, x_i and x_j independently fail.

Assumption 4. Any $y_k \in Y$ has two states: up (working) and down (failed), where y_k is up iff all elements of $\varphi(y_k)$ are working.

Assumption 5. For any $A_1 \in \rho(Y)$, $A_2 \in \rho(Y)$, if $A_1 \subseteq A_2$, then $C_0(A_1) \leq C_0(A_2)$.

C. Reliability measure

The reliability measure $R(N)$ for model N rests upon the following definitions.

Definition 1: X_g is a subset of X in which:
Every element of X_g is working.
Every element of $X - X_g$ has failed.

Definition 2: N's capacity $C(N, X_g)$, which implies the capacity ensured by the working traffic paths, is $C(N, X_g) = C_0(\{y \mid y \in Y, \varphi(y) \subseteq X_g\})$.

Definition 3: The reliability measure for N is $R(N) = \Pr(C(N, X_g) \geq \alpha)$, where α is a specified positive real number.

The failure of an element of X alters the elements of X_g by Definition 1. This alteration of X_g changes the value of $C(N, X_g)$ according to Definition 2. This mechanism, together with Assumption 2, lead to the result that $C(N, X_g)$ fluctuates probabilistically.

If we assign α the value of $C(N, X_g)$ that leads to a sufficient customer-satisfaction threshold, then a value of $\Pr(C(N, X_g) \geq \alpha)$ near 1 implies high reliability in the sense that it stably achieves customer satisfaction. Therefore, $R(N) = \Pr(C(N, X_g) \geq \alpha)$ in Definition 3 is reasonable as a reliability measure.

D. Algorithm for reliability evaluation

Here, we explain the algorithm proposed in ref. [6] which is the fastest exact evaluation for $R(N)$.

Reduction

In Fig. 1, x_2 is not included in any traffic path; therefore, failure of x_2 never affects reliability. That is, even if we delete x_2, the value of $R(N)$ does not change (Reduction Rule 1). Moreover, for x_7 and x_8, in Fig. 1, the surviving traffic paths in the case of x_7 failing are the same as in the case of x_8 having failed. Therefore, even if we combine x_7 and x_8 into one physical element whose failure probability is $\Pr(x_7$ fails or x_8 fails), the value of $R(N)$ never changes (Reduction Rule 2).

Factoring

We define 'factoring model N by x' to mean generating model N_x and N_x^D from N, where N_x & N_x^D are defined below.

N_x: The traffic-path-based network model obtained from N by deleting $x \in X$ from X, and deleting x from $\varphi(y)$ for every $y \in Y$.

N_x^D: The traffic-path-based network model obtained from N by deleting traffic paths included in $\{y \mid y \in Y, x \in \varphi(y)\}$, deleting $x \in X$, and deleting x from $\varphi(y)$ for every $y \in Y$.

In N_x, x is deleted, but the traffic paths going through x are not deleted. Ref. [6] noted that this situation implies that x never fails.

Fig. 2 shows an example of factoring. The thick bold line implies the corresponding element is deleted but the traffic path going through this element is not deleted.

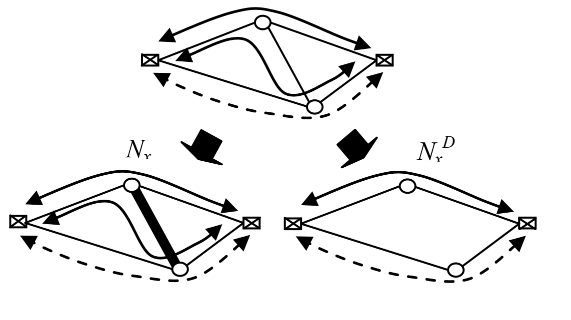

Fig. 2 Example of factoring.

Ref. [19] derived the following equation.

$$R(N) = p_i R(N_x) + q_i R(N_x^D)$$

From it, we can obtain the value of $R(N)$ if $R(N_x)$ and $R(N_x^D)$ are known. $R(N_x)$ and $R(N_x^D)$ can be obtained by factoring again after applying Reduction Rules 1 & 2.

Ref. [6] proposed an evaluation algorithm combining factoring and reductions with emphasizing that their proposed algorithm is faster at evaluating $R(N)$ for model N than the one in [5].

While the above emphasis, the tree in Fig. 3 grows exponentially as the model size becomes larger. Therefore, we need a more effective method to evaluate $R(N)$ for large N.

III. Proposal

Here, we propose a new method for evaluating $R(N)$. The key idea is using the BDD method, which is a relatively recent approach used for evaluating the reliabilities of other models [2][7]. This is the first use of this idea in a traffic-path-based network model.

A. BDD algorithm for $R(N)$ for model N

Fig. 3 is a binary tree expansion expressing the repetitions of factoring for N without reduction. These repetitions are executed until we need no further factoring (If Conditions 1, 2, and 3 in the algorithm of ref. [6] are satisfied then we can find no need for further factoring, where these conditions are same as BDD(N, α) shown in the later part in this section.)

In this figure, we find that $(N_{x_3})_{x_4}{}^D$, $(N_{x_3}{}^D)_{x_4}$, and $(N_{x_3}{}^D)_{x_4}{}^D$ are the same models, because the bold line implies the deletion of the corresponding element of X, as described in the explanation of factoring in Subsection D of Section II.

Therefore, we do not need to evaluate $R((N_{x_3}{}^D)_{x_4})$ and $R((N_{x_3}{}^D)_{x_4}{}^D)$ if we have already evaluated $R((N_{x_3})_{x_4}{}^D)$ in this binary tree and memorized the result. That is, Fig. 3 can be simplified to Fig. 4.

The improved algorithm using the above idea for $R(N)$ is shown below, where Q is the set of two-dimensional vectors (Q_1, Q_2) in which Q_1 expresses the model appearing in the binary tree and Q_2 expresses the value of $R(Q_1)$. Before starting the improved algorithm, Q is set to (ϕ, ϕ). Bold letters in the improved algorithm below indicate the primary differences from ref. [6].

<u>BDD(N, α)</u>

Input N, α

Output $R(N)$

Step 1. End BDD( ) after outputting the value of $R(N)$ if N is found to be equivalent to either model in the first column of vector in Q, where $R(N)$ is obtained from the second column.

Step 2. If N satisfies any of the following conditions;

Condition 1. Y includes a single traffic path (denoted by y).

Condition 2. $C(N, \varphi(Y)) < \alpha$

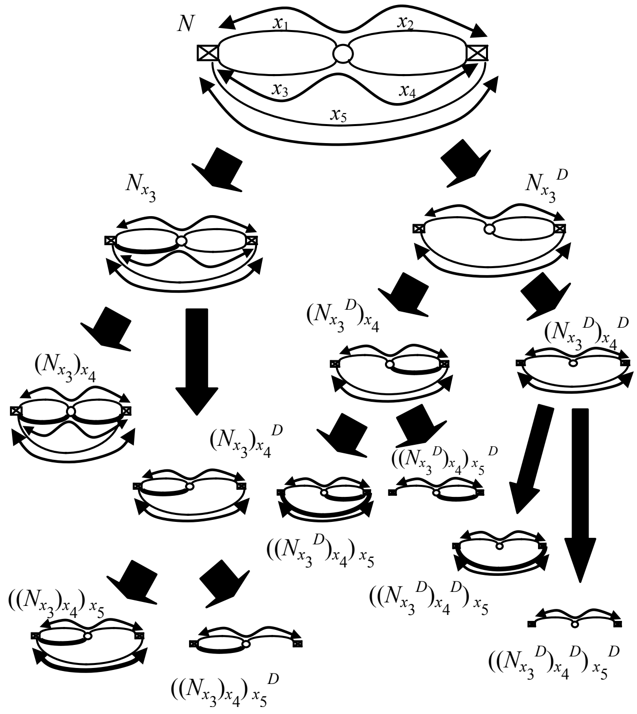

Fig. 3 A binary tree for factorings.

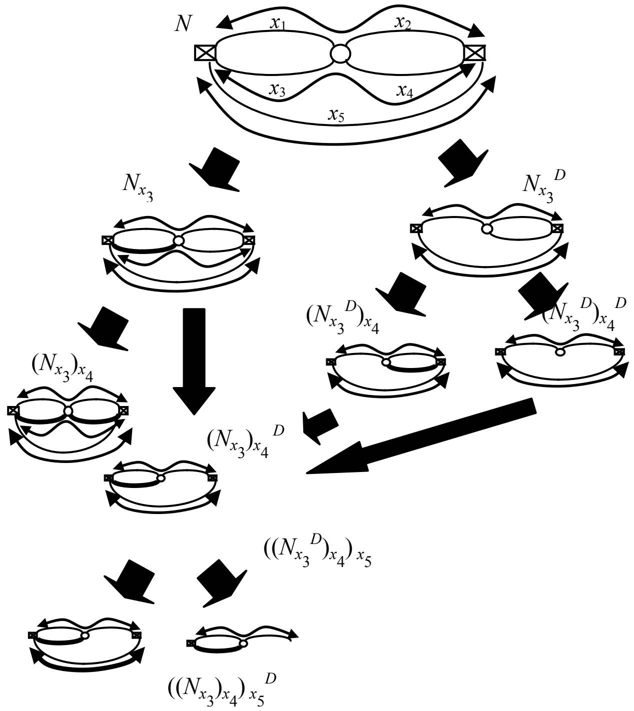

Fig. 4 BDD for N.

© Copyright IEICE – The 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) 2019

Condition 3. $C_0(A_0) \geq \alpha$, if A_0 is the set of traffic paths that are empty sets.

then end BDD( ) after outputting the value of $R(N)$ as follows, while storing $(N, R(N))$ in Q.

If Condition 1 is true and $C_0(\{y\}) \geq \alpha$, then $R(N)$ is the probability of all elements in $\varphi(y)$ working.

If Condition 1 is true and $C_0(y) < \alpha$, then $R(N) = 0$.

If Condition 2 is true, then $R(N) = 0$.

If Condition 3 is true, then $R(N) = 1$.

Step 3. Select an element of X, denoted by x_i, and execute

$$BDD(N, \alpha) = p_i BDD(N_x, \alpha) + q_i BDD(N_x, \alpha).$$

Add $(N, BDD(N, \alpha))$ to Q.

IV. NUMERICAL EXPERIMENTS

A. Preparation

A graph is defined as a set of nodes and links, where link connects a pair of nodes. Suppose that a link e connects a pair of nodes i & j; then we write $e = (i, j)$. We call i & j the end nodes of link e. The path between two nodes i_1 & i_m is defined to be an alternating sequence of nodes and links: i_1 - (i_1, i_2) - i_2 – (i_2, i_3) - … - (i_{m-1}, i_m) - i_m. Suppose a path includes L links. Then L is called the length of this path.

If every pair of nodes (a, b) of a graph has a single link whose end nodes are a & b, then the graph is called complete. If a complete graph has n nodes, it is called an n-complete graph. For example, a 5-complete graph is illustrated in Fig. 5.

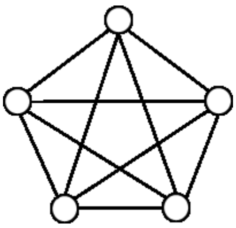

Fig. 5 A 5-complete graph.

B. Tarrgets of the evaluation

In our numerical experiment, X is the set of nodes and links of complete graphs, where we assume that both nodes and links can fail with the same probability, $q_i = 0.00001$. We dare to select such small value for q_i because its computation becomes difficult if we input such small value. That is, if we have no problem for such small value in our numerical experiment then other bigger values will also show no problem.

We focus on complete graphs because they will show the worst case computation results compared with other graphs. If we can find reasonably effective computation speed for them then reliabilities of other graphs can be computed with more reasonable computation times.

Traffic paths are selected from paths between nodes s & t so that their lengths should be less than 4. The length '4' is selected because if it includes paths with long lengths then the model becomes near to graph models.

$C_0(\ )$ is defined as the number of working traffic paths. The threshold value α is defined to be the half the number of traffic paths.

C. Environment

The environment for our experiments was as follows.

Software language: C#   OS: Windows 7 Professional 64bit
CPU: Intel®Core™i7-3770   Memory size: 4.00Gb

D. Evaluation results

Table 1 shows the results of the evaluation.

Table 1. Evaluation results

n	Evaluation time by ref. [6]	Evaluation time by BDD( )	$R(N)$
6	0.017	0.029	0.99998000
7	0.123	0.064	0.99998000
8	3.966	0.288	0.99998009
9	124.948	1.975	0.99998008
10	69531.692	11.578	0.99998007

For $n = 6$, the previously proposed method is slightly faster than our method. This is because searching for the same models in BDD for N needs extra computations. However, our method becomes dramatically faster than the previously proposed one as n becomes bigger.

That is, our method is surely faster than the previously proposed one when there are many physical elements and traffic paths.

V. CONCLUSION

This paper proposed a new approach for evaluating the reliability of traffic-path-based network models. This approach is based on the binary decision diagram method.

The numerical experimental results show that our proposal is faster than the existing method at evaluating the reliability of large traffic-path-based models.

REFERENCES

[1] S, Y. Kuo, F. M. Yeh, and H. Y. Lin, "Efficient and exact reliability evaluation for networks with imperfect vertices", IEEE Trans. Reliability, vol. 56, no. 2, pp.288-300, 2007.

[2] G. Hardy, C. Lucet, N. Limnios, "K-terminal network reliability measures with binary decision diagram", IEEE Trans. Reliability, vol. R-56, no.3, pp. 506-515, 2007.

[3] J. Malinowski, " A new efficient algorithm for generating all minimal tie-sets connecting selected nodes in a mesh-structured network", IEEE Trans. Reliability, vol. 59, no. 1, pp. 203-211, 2010.

[4] M. Hayashi and T. Abe, "Evaluating reliability of telecommunications networks using traffic path information", IEEE Trans. Reliability, vol. R-57, pp. 283-294, 2008.

[5] M. Hayashi and T. Abe, "Efficient reliability approximation method for traffic-path-based network", IEEE Trans. Reliability, vol. R-60, no. 2, pp. 460-469, 2011.

[6] M. Hayashi and T. Abe, "Evaluating reliability of telecommunications networks with traffic path restrictions", IEICE Trans. A, vol. J94-A, no. 8, pp. 629-638, 2011.

[7] M. Kvassay, E. Zaitseva, V. Levashenko, and J. Kostolny, "Binary decision diagram in reliability analysis of standard system strucure", Internatial Conference on Information and Digital Technologies (DT), pp. 164-172, 2016.