Ning HUANG
Dong HOU
Yang CHEN
Liudong XING
Rui KANG

A NETWORK RELIABILITY EVALUATION METHOD BASED ON APPLICATIONS AND TOPOLOGICAL STRUCTURE

METODA OCENY NIEZAWODNOŚCI SIECI OPARTA NA APLIKACJACH I STRUKTURZE TOPOLOGICZNEJ

Applications play an important role in the reliability evaluation of communication networks. In other words, the reliability of a network can be totally different when different applications are considered for the same network. However existing reliability evaluation methods, which are mostly based on the graph theory, give no or little consideration to applications. This paper proposes a concept of network application reliability and a Markov-based method for analyzing the proposed network application reliability measure. Furthermore, based on the reliability of each individual application, a method is proposed to evaluate the overall network reliability that incorporates effects of different applications running on the network. Both a case study and experiments are performed to illustrate the proposed concept and methods.

Keywords: *application, Markov model, network, reliability.*

Aplikacje odgrywają ważną rolę w ocenie niezawodności sieci komunikacyjnych. Innymi słowy, niezawodność sieci może być całkowicie różna dla różnych aplikacji tej samej sieci. Niestety, istniejące metody oceny niezawodności, w większości oparte na teorii grafów, poświęcają niewiele lub nie poświęcają wcale uwagi aplikacjom. W niniejszym artykule przedstawiono koncepcję niezawodności aplikacji sieciowych oraz opartą na modelu Markowa metodę analizy proponowanej miary niezawodności aplikacji sieciowych. Ponadto, na podstawie niezawodności poszczególnych aplikacji, zaproponowano metodę oceny ogólnej niezawodności sieci, która łączy efekty różnych aplikacji działających w danej sieci. Zaproponowaną koncepcję i metody omówiono na podstawie studium przypadku oraz badań eksperymentalnych.

Słowa kluczowe: *aplikacja, model Markowa, sieć, niezawodność.*

1. Introduction

Reliability analysis has become an essential step in the design, operation, and tuning of network systems [4, 24]. Considerable research efforts have been expended in the network reliability analysis. As a pioneer, Lee first defined and evaluated the network reliability mainly based on network connectivity [6]. Following the similar idea, a series of network reliability evaluation algorithms and optimization methods have been proposed [1, 3, 7, 8, 9, 12, 13, 19, 20, 21, 23, 28, 30, 32]. All these studies are based on the graph theory and the network topology. Among these works, synthesis evaluation methods are especially discussed in [1, 7, 8, 28, 30]. Later on other more advanced synthesis evaluation methods have been proposed, including but not limited to AHP (Analytic Hierarchy Process) [14, 27], fuzzy reliability evaluation [17], and ANN (Artificial Neural Network) [5]. Since 1980s, network congestion and traffic delay have become noticeable factors in the network reliability research. For example, Barberis and Park investigated network availability considering throughput and delay [10, 25]. Tao and Chen considered routing dynamics and congestion into the network reliability computation [33]. In general, the existing network reliability research can be classified into two types [29]: *inherent reliability* considering topology connectivity and *applicable reliability* considering network traffic. The former focuses on topology structure, and has been analyzed using probability theory and graph theory. The latter focuses on how the network works and what is in the network, and examines performance reliability of the network [11, 15, 22].

Those research works on network reliability, however, gave little or no consideration to the effects of applications, though the network reliability and performance can be different when different applications run on the network. Recently studies on application-layer network performance testing [18, 26] have started. The performance of application layer is quite different from and not directly related to the performance of the other layers in the OSI model. And end users are usually concerned with the performance of specific applications [2, 31]. It is worth noting that the Internet Engineering Task Force (IETF) proposed a performance testing methodology and some metrics on the application layer in RFC 3511 [16]. And the influence of applications for network reliability has also been noticed in [34, 35].

To the best of our knowledge, there is no work considering various applications and incorporating the effects of them in the evaluation of the network reliability. In this paper, a concept of network application reliability is proposed and a Markov-based method for analyzing the proposed application reliability measure is discussed. Furthermore, based on the reliability of each individual application, a method is proposed to evaluate the overall network reliability that considers the effects of different applications running on the network. Both a case study and experiments are performed to illustrate the proposed concept and methods.

2. Concept and model

The following concepts are defined and used in the latter discussions:

A network: is a group of hardware devices and services. It has the transportation ability to support applications for users.

A service: is a function that a network provides to users. Usually a function is supported by a software system or a group of cooperating software systems.

An application: is the usage of services by a group of users with some demanded performance requirements.

Application profile: represents the information of an application, including information of the users involved and a set of operation probabilities of the application;

Usage profile: contains a set of application profiles and their occurrence probabilities.

The proposed application-centric model for communication network reliability is shown in fig.1.

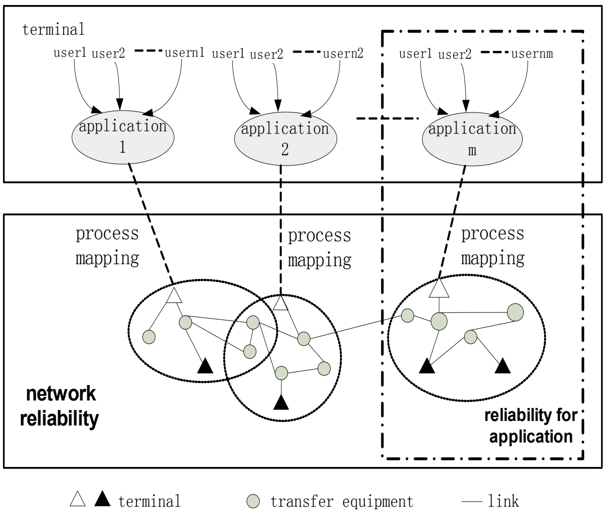

Fig.1. Network reliability model

Let R represent the overall network reliability, and R_i represent reliability for application A_i. R is a function of R_i and usage-profile. Namely R is a function of reliabilities for all applications, and the relationship among all the applications in the network. Let H_i be the set of hardware that application A_i involves. $\{Feature_j\}_i$ means a set of reliability of each element in H_i. Before evaluating the application reliability, each application should be mapped to its topology. The main idea of this mapping is to separate devices into groups according to the usage of the application, as illustrated in fig.1. We use *topo* log y_i to indicate the mapped topology of H_i, namely how the hardware components involved in A_i are connected. $Deploy_i$ represents how application A_i is deployed in the network. *App - profile*$_i$ shows the application profile of application A_i. The application reliability R_i is a function of $\{Feature_j\}_i$, *topo* log y_i, $deploy_i$ and *App - profile*$_i$.

3. Proposed algorithms

Algorithms for computing the application reliability and the overall network reliability are explained in this section. Two assumptions are made in the proposed algorithms: 1) All the network components (nodes or links) fail statistically-independently; and 2) The transfer of data flow in the network is a Markov process, meaning that the determination of the next node to transfer the flow depends on the present node, not the past path.

As briefed in Section 2, R_i is a function of $\{Feature_j\}_i$, *topo* log y_i, $deploy_i$, and *app - profile*$_i$, and R is the function of $\{R_i\}$ and usage-profile. The substances of the parameters in this algorithm are:

1) $Feature_j\}_i$: reliabilities of the components.
2) *Topo* log y_i: the transfer matrix.
3) $Deploy_i$: mapping to the network components according to the deployment of the services.
4) *App - profile*$_i$: the transfer probability of the application when it is running.
5) *Usage - profile*: the number and importance of the applications.

The general algorithm can be described as the following process:

Step 1: prepare $\{Feature_j\}_i$ and *Usage - profile*: Analyze the network, and prepare the static parameters including the reliabilities of nodes and links from the history usage data (they usually can be obtained from devices providers). Assign the weight for each application according to its importance in the network.

Step 2: analyze applications on the network. For each application A_i, determine its concerned services as a set S_i, and its process (data flow) for the network reliability.

Step 3: analyze $deploy_i$ for each application A_i. Based on the information obtained in step2, analyze where each service is installed and what nodes and links its process concerns. Because the data flow has direction, every application A_i is mapped to two diagrams, request diagram and response diagram, corresponding to request data flow and response data flow, respectively.

Step 4: analyze *App - profile*$_i$. Acquire the multi-branch transfer probabilities of the nodes for each application A_i using history data or statistic methods for both request diagram and response diagram.

The transfer probability of a link is computed as the ratio of the size of the data flow through the link to the total size of data flows through all the links involved in the application. For example, a node used in a specific application connects with three links which are named as *a*, *b*, and *c*, respectively. The sizes of the data flows going through the three links are respectively 20KB, 30KB and 50KB. Thus the transfer probabilities of the three links are respectively 0.2, 0.3, and 0.5.

Step 5: calculate reliability R_i for each individual application A_i using the method presented in Section 3.1.

Step 6: evaluate the overall network reliability using the method described in Section 3.2.

3.1. Reliability evaluation for an application

Let H_i^s and H_i^t respectively represent the set of nodes for request diagram and response diagram of application A_i, R_i^s

and R_i^t respectively represent reliability corresponding request diagram and response diagram. There are six steps to compute the reliability of application A_i.

Step 1: obtain the transfer matrix Q based on the request diagram of application A_i.

Let n_i and n_j represent network nodes, namely $n_i, n_j \in H_i^s$. Represent a link from n_i to n_j with l_{ij}, the reliability of n_i with N_i, the reliability of l_{ij} with L_{ij}, and the transfer probability from n_i to n_j with P_{ij}. We regard every node in the network as a state of the Markov model. The model also includes states C and F that represent the application request is completed successfully (N_C=1) and the request is failed (N_F=1) respectively. Thus the complete state space for the Markov model is $\{n_1,n_2,\ldots,n_n,C,F\}$. The state transition matrix is named T:

$$T = \begin{array}{c} \\ n_1 \\ \vdots \\ n_i \\ \vdots \\ n_{n-1} \\ n_n \\ C \\ F \end{array} \begin{array}{c} \begin{array}{cccccccc} n_1 & n_2 & \cdots & n_j & \cdots & n_n & C & F \end{array} \\ \left[\begin{array}{cccccccc} 0 & N_1L_{12}P_{12} & \cdots & N_1L_{1j}P_{1j} & \cdots & N_1L_{1n}P_{1n} & N_1L_{1C}P_{1C} & 1-F_1 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & \vdots \\ 0 & N_iL_{i2}P_{i2} & \cdots & N_iL_{ij}P_{ij} & \cdots & N_iL_{in}P_{in} & N_iL_{iC}P_{iC} & 1-F_i \\ \vdots & \vdots & & \vdots & & \vdots & \vdots & \vdots \\ 0 & N_{n-1}L_{(n-1)2}P_{(n-1)2} & \cdots & N_{n-1}L_{(n-1)j}P_{(n-1)j} & \cdots & N_{n-1}L_{(n-1)n}P_{(n-1)n} & N_{n-1}L_{(n-1)C}P_{(n-1)C} & 1-F_{n-1} \\ 0 & N_nL_{n2}P_{n2} & \cdots & N_nL_{nj}P_{nj} & \cdots & 0 & N_nL_{nC}P_{nC} & 1-F_n \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \end{array}\right] \end{array} \quad (1)$$

Where,

$$F_i = \sum_{j=2}^{n} N_iL_{ij}P_{ij} + N_iL_{iC}P_{iC}, L_{iC} = 1, P_{iC} = 0 or 1 \quad (2)$$

Every element $T(a,b)$ in matrix T represents the probability of successful transfer of a flow from node n_a to node n_b. The value of $T(a,b)$ is the product of N_a, L_{ab} and P_{ab}. For instance, the element $T(i,2)$ in row n_i and column n_2 is the product of the reliability of n_i, the reliability of l_{i2} and the transfer probability from n_i to n_2, namely the product of N_i, L_{i2} and P_{i2}. n_1 represents the requester of the application whose in-degree is 0. C represents the completion state and F represents the failure state. Both of them are absorbing states with out-degrees of 0. Therefore, elements in the first column and the last two rows in T are 0.

The matrix Q is obtained by removing the row and column of F from matrix T.

$$Q = \begin{array}{c} \\ n_1 \\ \vdots \\ n_i \\ \vdots \\ n_{n-1} \\ n_n \\ C \end{array} \begin{array}{c} \begin{array}{ccccccc} n_1 & n_2 & \cdots & n_j & \cdots & n_n & C \end{array} \\ \left[\begin{array}{ccccccc} 0 & N_1L_{12}P_{12} & \cdots & N_1L_{1j}P_{1j} & \cdots & N_1L_{1n}P_{1n} & N_1L_{1C}P_{1C} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & N_iL_{i2}P_{i2} & \cdots & N_iL_{ij}P_{ij} & \cdots & N_iL_{in}P_{in} & N_iL_{iC}P_{iC} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & N_{n-1}L_{(n-1)2}P_{(n-1)2} & \cdots & N_{n-1}L_{(n-1)j}P_{(n-1)j} & \cdots & N_{n-1}L_{(n-1)n}P_{(n-1)n} & N_{n-1}L_{(n-1)C}P_{(n-1)C} \\ 0 & N_nL_{n2}P_{n2} & \cdots & N_nL_{nj}P_{nj} & \cdots & 0 & N_nL_{nC}P_{nC} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \end{array}\right] \end{array} \quad (3)$$

Step 2: derive matrix W, and calculate the determinant of W denoted as $|W|$.

For any integer m (m>0), $Q^m(i,j)$ is the probability that the data packet is transferred from n_i to n_j within m steps. It is supposed that S is a matrix with the order of n+1, and:

$$S = I + Q + Q^2 + \cdots = \sum_{k=0}^{\infty} Q^k \quad (4)$$

where, I is an identity matrix.

The application reliability is thus the transfer probability from n_1 to C, that is, $R_i^s = S(n_1, C)$. When $W = I - Q$, we have:

$$S = W^{-1} = (I - Q)^{-1} \quad (5)$$

Then the value of $|W|$ can be computed.

Step 3: remove the first column and the last row of W, name the remaining matrix as M, and calculate $|M|$.

Step 4: calculate R_i^s as the reliability for request diagram of application A_i with the formula:

$$R_i^s = (-1)^n \frac{|M|}{|W|} \quad (6)$$

where, n is the number of nodes in the application request diagram.

Proof:

$$R_i^s = S(n_1, C) = W^{-1}(n_1, C) = \frac{W^*(n_1, C)}{|W|} = \frac{(-1)^{n+1+1}|M|}{|W|} = (-1)^n \frac{|M|}{|W|} \quad (7)$$

where, W^* is the adjoint matrix of W.

Step 5: using the above similar steps, calculate R_i^t based on the response diagram for application A_i.

Step6: computed reliability for application A_i using the following formula:

$$R_i = R_i^s \times R_i^t \quad (8)$$

3.2. Reliability evaluation for the entire network

There is typically more than one application existing in the network. Therefore a method is needed to integrate the single application reliabilities to obtain the entire network reliability. In this work, the overall network reliability is evaluated as a weighted sum of reliabilities of all applications running on the network, as shown in (9).

$$R = \sum_{i=1}^{n} \omega_i R_i, \sum_{i=1}^{n} \omega_i = 1 \quad (9)$$

Where ω_i represents the weight of application A_i, which indicates the number of users or the significance of the application. Consider an example where there are three applications of three groups of users called Lan1, Lan2 and Lan3 with the same significance level. The topology reliabilities of these applications are respectively R_1 = 0.9, R_2 = 0.8, R_3 = 0.9, and the number of users of Lan1, Lan2 and Lan3 are respectively 6, 7 and 7. The weights of these applications are computed as ω_1= 6/(6+7+7)=0.3, ω_2=0.35, ω_3=0.35. Thus, the overall network reliability is $R = \sum_{i=1}^{3} \omega_i R_i = 0.865$.

4. Case study

In this section, a case study is performed to show how a network reliability can be evaluated using the method described in Section 3.

4.1. System description

Figure 2 illustrates a small campus network with teaching VOD (Video on Demand) applications running on it. Reliabilities of nodes and links in this network are given in table 1 and table 2, respectively. Lan1, Lan2 and Lan3 are three different groups of users. There are 10 users in Lan1, 15 in Lan2, and 25 in Lan3. Lan 1 is a LAN of student dormitories and faculty apartments; Lan 2 is a LAN of the teaching zone, and Lan 3 is a LAN of the teaching showcase area.

Service1 and Service2 are grouped together to support a VOD providing application where users can watch part of the teaching videos and TVs (referred to as application1 hereafter), and they are installed separately on Server1 and Server3. Service3 itself also supports the same VOD providing application as a main server for all the video sources (application2) and it is installed on Server2. Service4 supports a HTTP application where users access to the Internet or other communication networks (application3), and is installed on Server2 too. These three applications run on the network: users can visit application1 through Lan1, application2 through Lan2, and application3 through Lan3.

4.2. Network Reliability Evaluation

Using the method of Section 3, the reliability of this example network reliability can be evaluated using the following steps:

Step 1: prepare data, including reliabilities of the nodes and links, shown in tables 1 &2.

Step 2: analyze applications. For example, for application1, users request Service1 (on Server1) for a special video by a browser. If this video can be provided by Service1, it can be downloaded by the users. Otherwise, the request will be transferred to Service2 (on Server3) to find the video.

For application1, S_1={browser, Service1, Service2}. In this step, only services are analyzed without consideration of hardware, namely, these services can be installed on different servers involving different transform devices.

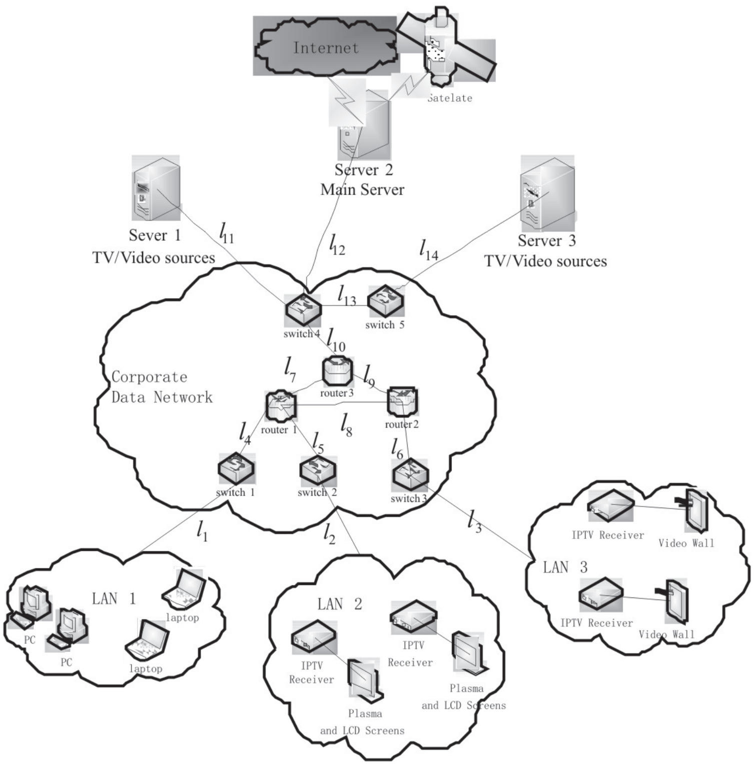

Fig. 2 Network structure of the case

Tab. 1. Reliability of the Nodes

Node	R_i	Node	R_i
lan1	1	router2	0.97
lan2	1	router3	0.99
lan3	1	switch4	0.98
switch1	0.98	switch5	0.98

Tab. 2. Reliability of the Links

Link	L_i	Link	L_i
l_1	0.99	l_8	0.98
l_2	0.98	l_9	1
l_3	0.98	l_{10}	0.99
l_4	1	l_{11}	0.99

Step 3: figure out the request diagram for applicaiton1. Based on S_1 obtained in Step2, servers, switches, and routers are figured out based on the actual network configuration, shown in figure 3. Note that if Service1 and Service2 are installed on different servers or the different routing rules are involved, then the diagram will be different.

In this case, request diagram concerns all the devices to support application1. But if the routing rules in Router1 are changed to deliver its data packages only to Router3, then the request diagram will be changed to another one without Router2.

Step 4: establish the transfer probabilities of the multi-branch nodes for request diagram for application1.

In this example system, the requests of application1 are sent to server1 with a probability of 0.8, and to server 3 with a probability of 0.2 (when the object file is not found on server1). That is, the transfer probabilities of l_{11} and l_{13} are 0.8 and 0.2, respectively. Similarly, the transfer probabilities of l_7 and l_8 are 0.6 and 0.4, respectively. For the nodes with only one outgoing link, the transfer probability of the link is simply 1.

Step 5: calculate R_1^s and R_1^t for application1. Here the evaluation of R_1^s is explained in detail:

1) Add a state C for application1 indicating that this application request is completed successfully. Derive matrix Q_1 from the reliabilities of correlative nodes and links as well as the transfer probabilities.
2) Compute the matrix $W_1=I-Q_1$, and the value of its determinant $|W_1|$. W_1 is an upper triangular matrix here, and $|W_1|=1$.
3) Remove the first column and the last row from W_1 to get a new matrix M_1, then obtain the result $|M_1|\approx-0.86852$.
4) The request reliability of application1 can be calculated as:

$$R_1^s = (-1)^n \frac{|M_1|}{|W_1|} \approx (-1)*\frac{-0.86852}{1} = 0.86852$$

where, n represents the number of nodes in the topology of figure 3, which is 9.

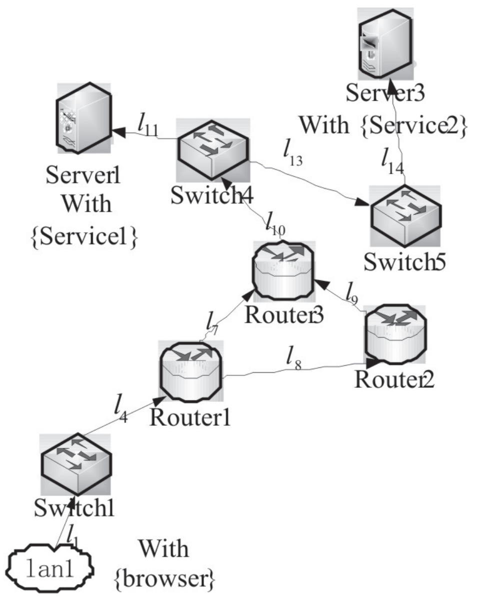

Fig. 3 Request Diagram of application1

Calculate reliability R_1^t in the similar way. In this example, response diagram is the same as the request diagram with opposite directions. So we have $R_1^t = R_1^s = 0.86852$. Thus, $R_1 = R_1^s \times R_1^t = 0.86852^2 = 0.75433$.

Step 6: Similarly, following step 2 to step 5, reliabilities of application2 and application3 can be evaluated as: $R_2\approx 0.68800$, $R_3\approx 0.69499$.

Step 7: calculate the network reliability as a weighted sum of all the single application reliabilities. The weights of the three applications are calculated as the proportion of their users. Thus the entire network application is calculated as: $R = \sum_{i=1}^{3} \omega_i R_i = 0.2*0.75433 + 0.3*0.68800 + 0.5*0.69499 = 0.704761$.

5. Experiments and analysis

Further experiments are performed on the example network under different conditions to study the effects of component reliability and applications on the network reliability.

5.1. Experiment 1: Influence of component reliability on network reliability

Fig. 4. shows the change of the application reliability R_1, R_2 and R_3, and network reliability R, when the reliability value of switch4, N_{s4} changes. Similarly, fig. 5. shows the effect of the change of reliability of switch5 N_{s5} on the reliabilities of single applications and the entire network.

Based on fig. 4 and fig. 5, we can see that R_1, R_2, R_3, and R decrease as N_{s4} or N_{s5} decreases. In addition, switch4 is more important than switch5 to application 1 and application 2, because R_1 and R_2 reduce more rapidly in fig. 4 than in fig. 5. As shown through this example, our algorithm can facilitate the study of sensitivity or importance of different components to the network reliabilities.

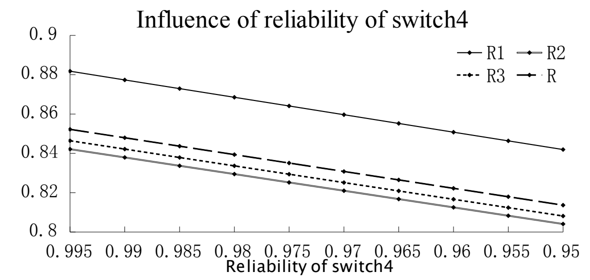

Fig. 4 Reliabilities decrease as the reliability of switch4 is reduced

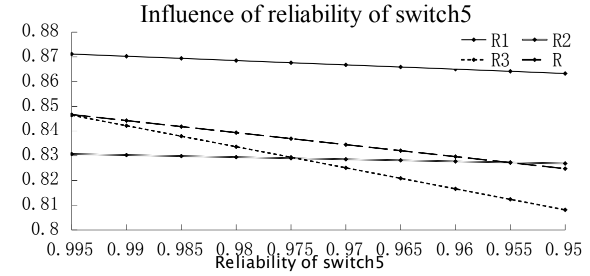

Fig. 5 Reliabilities decrease as the reliability of switch5 is reduced

5.2. Experiment 2: Influence of applications on network reliability

Experiments are performed on the example network to study the effects of applications on the network reliability.

When the proportion of the requests sent to Server1 and Server3 in applications changes, the changes of network reliability are shown in figure. 6. In particular, as the portion of requests sent to Server 1 decreases and portion sent to Server 3 increases, both the reliability of application1R_1, and the overall network reliability R decrease. This is reasonable because the branch to Server1 has a greater reliability than the branch to Server3.

The influence of component reliability on the overall network reliability has been studied and widely acknowledged. Meanwhile, the reliability of a network can be totally different when different applications are considered for the same network. Experiments performed in Sections 5.1 and 5.2 show that our evaluation method reflects the influence of not only components but also applications on the network reliability. This research is our first step for studying application-oriented reliability for communication networks with deterministic routings. It has provided another view of network reliability, which can reflect the users' requirements better.

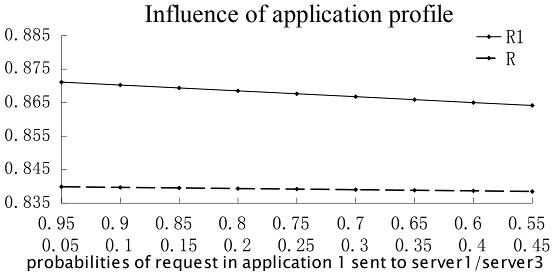

Fig. 6. The influence of applications on network reliability

6. Conclusions and future work

Traditional network reliability algorithms mainly focus on network topology/connectivity while giving little or no consideration to applications running on the network. Thus results obtained using the traditional methods are not convincing enough for practical projects in enterprises because applications can affect the performance/reliability of a network greatly. A new application-centric network reliability concept and corresponding evaluation algorithm have been proposed in this paper. As shown through the case study and experiments, the algorithm considers the effects of both component reliabilities and applications in the network reliability evaluation.

Our future work will focus on (1) how to classify applications to reduce the computational overhead when the number of service is large; (2) how to optimize the algorithm to avoid the computational complexity caused by the excessive matrix order when the number of nodes related to a specific application is enormous; and (3) how to abstract more useful information about applications and components and incorporate it into the algorithm.

7. References

1. A. R. Abdelaziz. A New Approach to Power Network Synthesis. Proc. Of the 8th International Conference on Harmonics and Quality of Power, 1998:495.
2. A. Avritzer, E. J. Weyuker. The Role of Modeling in the Performance Testing of E-Commerce Applications. IEEE Transactions on Software Engineering, 2004, 30(12): 1072-1083.
3. A. M. Shooman. Algorithms for Network Reliability and Connection Availability Analysis. IEEE Transactions on Reliability, 1995, 44(10): 309-333.
4. A. P. Snow. Network Reliability: The Concurrent Challenges of Innovation, Competition and Complexity. IEEE Transactions on Reliability, 2001, 50(1): 3 8-40.
5. Chat Srivaree-ratana, Abdullah Konak and Alice E. Smith. Estimation of all-terminal network reliability using an artificial neural network. Computers & Operations Research, 2002, 29(7): 849-868.
6. C. Y. Lee. Analysis of Switching Networks. Bell Syst. Tech.J., 1955, 34: 1287-1315.
7. F. T. Boesch. Synthesis of Reliable Networks-A Survey. IEEE Transactions on Reliability, 1986, 35(3): 240-246.
8. F. T. Boesch. On Unreliability Polynomials and Graph Connectivity in Reliable Network Synthesis. Journal of Graph Theory, 1986, 10(3): 339-352.
9. Fu-Min Yeh, Hung-Yau Lin, Sy-Yen Kuo. Analyzing Network Reliability With Imperfect Nodes Using OBDD. Proc. of the 2002 Pacific Rim International Symposium on Dependable Computing, Tsukuba, Japan, 2002: 89-96.
10. G. Barberis and U. Mazzei. Traffic-based Criteria for Reliability and Availability Analysis of Computer Networks. Proc. IEEE ICC'77, Chicago, IL, Vol.1 session 22.6, June 1977.
11. Gao Zhuqing, Fan Hehong, Zhang Mingde, Sun Xiaohan. Connectivity Reliability Evaluation for Optical Layer in IP Over WDM Networks Based on Minimal Path Sets. Proceedings of the 7th International Conference of Advanced Communication Technology, Phoenix Park, Korea, 2005: 131-136.
12. Ghasemzadeh M., Meinel C., Khanji S.. K-terminal Network Reliability Evaluation Using Binary Decision Diagram. 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Umayyad Palace, Damascus, Syria, 2008: 1-5.
13. Héctor Cancela, Mohamed El Khadiri. The Recursive Variance-Reduction Simulation Algorithm for Network Reliability Evaluation. IEEE Transactions on Reliability, 2003, 52(2): 207-212.
14. Hossam Mahmoud Ahmad Fahmy. Reliability Evaluation in Distributed Computing Environments Using the AHP. Computer Networks, 2001, 36(5-6): 597-615.
15. Hui-Ling Liu, Shooman M.L.. Simulation of Computer Network Reliability with Congestion. Proceedings of Reliability and Maintainability Symposium, Washington D.C., USA, 1999: 208-213.

16. B. Hickman, D. Newman, S. Tadjudin, T. Martin, Benchmarking Methodology for Firewall Performance. RFC3511, April 2003
17. James Dunyak, Ihab W. Saad, Donald Wunsch. A Theory of Independent Fuzzy Probability for System Reliability. IEEE Transactions on Fuzzy Systems, 1999, 7(2): 286-294.
18. Jens Grabowski, Dieter Hogrefe, Gyorgy Rethy, Ina Schiferdecker, Anthony Wiles, Colin Willcock. An Introduction to the Testing and Test Control Notation (TTCN-3). Computer Networks, 2003, 42(3):3 75-403.
19. Jintao Xiong, Weibo Gong. A Novel Algorithm on Network Reliability Analysis. 10th International Conference on Telecommunications, Papeete, Tahiti (French Polynesia), 2003: 1407-1411.
20. J. M. Wilson. An Improved Minimizing Algorithm for Sum of Disjoint Products. IEEE Transactions on Reliability, 1990, 39(1): 42-45.
21. Johannes U.Herrmann and Sieteng Soh. A Memory Efficient Algorithm for Network Reliability. Proceedings of the 15th Asia-Pacific Conference on Communications (APCC 2009), Shanghai, China, 2009: 703-707.
22. Li Xiao, Nahrstedt.k. Reliability Models and Evaluation of Internal BGP Networks. INFOCOM 2004. Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, 2004: 1593-1604.
23. L. Xing. An Efficient Binary Decision Diagrams Based Approach for Network Reliability and Sensitivity Analysis. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2008, 38(1): 105-115.
24. M. O. Ball, C. J. Colbourn, J. S. Provan. Network Reliability. System Research Center, 1992.
25. Park Y J, Tanaka S. Reliability Evaluation of a Network with Delay. IEEE Trans. on Reliability, Vol. R-28, No. 4, 1979.
26. Peter James Schwaller, John Quillian Walker, Steven Thomas Joyce, Timothy Scott Huntley. Endpoint Node Systems Computer Program Products for Application Traffic Based Communications Network Performance Testing, U. S. Patent 6,061,725, May 9, 2000.
27. Qingyang Song, Jamalipour, A.. Network Selection in an Integrated Wireless LAN and UMTS Environment Using Mathematical Modeling and Computing Techniques. Wireless Communications, IEEE, 2005, 12(3): 42-48.
28. Rong-Hong Jan. Design of Reliable Networks. Computers & Operations Research, 1993, 20(1): 25-34.
29. Ruiying Li, Rui Kang, Ning Huang, Weiwei Chen, Yang Chen. A Practical Approach for Network Application Reliability Assessment. Eksploatacja i Niezawodnosc - Maintenance and Reliability, 2009, 43(4): 17-27.
30. Sawionek, B., Wojciechowski, J., Arabas, J.. Synthesis of Reliable Networks in the Presence of Line Failures. Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium, 2000: 649.
31. Siddhartha Dalal, Yu-Yun Ho, Ashish Jain, Allen McIntosh. Application Performance Assurance Using End-to-End User Level Monitoring. International Conference on Dependable Systems and Networks, 2002: 694-703.
32. Sy-Yen Kuo, Fu-Min Yeh and Hung-Yau Lin. Efficient and Exact Reliability Evaluation for Networks With Imperfect Vertices. IEEE Transactions on Reliability, 2007, 56(2): 288-300.
33. Tao Yu,Shanzhi Chen,Ming Ai. A Framework for Reliability Computation of the IP Network. Proceeding of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, China, 2007: 323-327.
34. Tatsuya Matsukwa, Hiroyuki Funakoshi. Analyzing Failure Frequency and Severity in Communication Networks, Reliability and Maintainability Symposium, San Jose, CA, 2010: 1-6.
35. Watanabe H, Kudo M, Kawanishi K, Yamasaki K. A Reliability Design Method for Private Networks. Reliability and Maintainability Symposium, 2003: 237-243.

Prof. Ning HUANG
Dong HOU
Yang CHEN
Prof. Rui KANG
School of Reliability and Systems Engineering
Beihang University, Beijing, P.R. China
e-mail: hn@buaa.edu.cn; houdong163@163.com
e-mail: chenyangjenny@126.com; kangrui@buaa.edu.cn

Dr. Liudong XING
Department of Electrical and Computer Engineering
University of Massachusetts, Dartmouth, MA 02747, USA
e-mail: lxing@umassd.edu