Journal of Web Engineering, Vol. 14, No.1&2 (2015) 136-150

PREVENTION OF FAULT PROPAGATION IN WEB SERVICE: A COMPLEX NETWORK APPROACH

YING LIU

Northeastern University, Shenyang, China

liuy@swc.neu.edu.cn

SHU MAO

Northeastern University,Shenyang, China

maoshu19900502@163.com

MINGWEI ZHANG

Northeastern University,Shenyang, China

zhangmw@swc.neu.edu.cn

GUOQI LIU

Northeastern University,Shenyang, China

liugq@swc.neu.edu.cn

ZHILIANG ZHU

Northeastern University, Shenyang, China

zzl@mail.neu.edu.cn

JINGDE CHENG

Saitama University, Saitama, Japan

cheng@ics.saitama-u.ac.jp

Received May 9, 2014
Revised October 2, 2014

How to prevent the fault propagation problems in Web Service has become an important issue. The recent research works mostly take some fault tolerance method in service based system. These methods detect or diagnose faults in the composition process, find the failure service, take tolerance action and recover the system. However, in the service oriented architecture, one service is shared by different service based systems. The fault tolerance method only considers from the view of one service user, and tolerance action not considering the whole network would change its load and even the global redistribution of loads over all of the services, trigger a cascade of overload, and result in service network paralysis. The research of

cascading failure in Complex Network provides a set of models to help study the above problems. Consequently, this paper proposes a new approach to deal with the fault propagation for Web Service from the view point of the whole service network, which could analyze its resistance influenced by the size of network, different types of attacks and load allocation strategies and prevent the disasters from happening. Firstly, it constructs a Web Service Complex Network (WSCN) composed of single service and their functional similarity. Then it models fault propagation based on WSCN, and simulates the propagation process by analyzing WSCN performance under small attack, large attack, random attack and calculated attack. When fault happens in WSCN, our method uses weight-based and spare-load-based load allocation methods of failed service to compare their influences on the whole network. The experimental results show that when fault happens in WSCN, the network has better resistance for small scale failure than big scale one, and resists stronger for random attack than deliberate one; when the service failure happens, the remaining space based load allocation strategy on it has higher robustness than weight based one. The simulation of fault propagation for Web Service could set example for preventing and reducing probabilities of collapse in the service network.

Key words: Web Service Complex Network, invocable relation between services, fault propagation, loads allocation strategy
Communicated by: B. White & E.-P. Lim

1 Introduction

With the increasing development of network applications, the information processing mode changes from centralized to distributed treatment gradually and Web Service emerges[1]. Web Service has the characteristics[2] of Encapsulation, loose coupling, self-describing, interoperability and universality, and therefore it is applied in all kinds of aspects more and more. As Web services are often long running and cross administrative boundaries, service based system may encounter various faults during the execution. When fault happened in the system, the failed service would be replaced by another one. These repairing actions would change the balance of flows, leads to a global redistribution of loads over all of the services, and even trigger a cascade of overload failures such as paralysis of the power system which was happened in USA in 2003[17]. How to replace and distribute the load of the failed service and prevent the fault propagation happening in service becomes an important issue.

However, the recent research works focus on fault tolerance of service based system including fault diagnosis[5,8-9,20] and recovery[6-7]. Yan et al [5] propose a model-based approach to diagnose the faults in a Web service-composed business process. Zhu et al [8] propose an execution flow model for service composition, and search the fault transmission flow path consisted by all potential faults. Based on the execution result, the original failure along the path is checked and identified by analyzing the status of every relate service. Friedrich et al [9] envision a different approach to exception handling in service-based system. In his approach, it exploits the causes of the process faults and derives the repair strategies from the structure of process. Liu et al [20] propose a framework for fault-tolerant composition of transactional Web services.

The above research works only resolve faults problems in service based system from the view of service user. They detect or diagnose faults in the composition process, find the failure service, and take tolerance method to recover the system. But they neglect the fault propagation caused by the continuous influence of the replacing action[3] in Web Service. For example, some of the replacing mechanisms take QoS into consideration, and select the service of highest QoS as the replacing service[12-13]. If more and more repairing actions choose this service at the same time, its load will be increased and then easily failed. This situation may lead to cascading failure of the replacing service and other related services,

and even cause the collapse in the service network. On one hand, we need tolerance method to resolve the existing faults for Web Service. On the other hand, we need effective method to prevent faults and fault propagation. Consequently, it is important to consider cascading failures on Web Service in order to better understand and control various cascading-failure-induced disasters. The research of cascading failure in Complex Network[10-11] provides a set of models to help study the above problems, such as the sand-pile model[14], the global load based cascading model GLBCM[15], and the fiber bundle model FBM[16].

In this paper, by referring the cascading models in Complex Network, we proposed a Complex Network approach for prevention of fault propagation in Web Service. Firstly, we constructed a Web Service Complex Network (WSCN for short) by setting Web Service as node and their relations as arc and defines service cascading failure model similar in Complex Network. Secondly, we analyzed the resistance of WSCN influenced by the size of network, different types of attacks and load allocation strategies. We simulated dynamic evolving process of WSCN when cascading failure was happening, and designed small attack, large attack, random attack and calculated attack on the network to compare the network performance under different attacks. Thirdly, we replaced the failed service node by its brother nodes in WSCN and used weight-based and spare-load-based load distribution method on it separately to analyze their different influence. The approach proposed in this paper dealt with the failed service by distributing its load to its related services based on WSCN, and it help us to solve the fault in service based system from the view of a whole network and prevent the happening of cascading failure. The result of experiments showed that the simulation could set example for preventing and reducing probabilities of collapse in the service network.

The rest of this paper is organized as follows: Section 2 describes the complex network model based on services' invocable relations. Section 3 discuses the cascading failure model. Section 4 shows the experiment results on networks of different size. Section 5 discusses the results of the experiments. Conclusions and Future Works are given in Section 6.

2 Web Service Complex Network Model

Through investigating the existing research work on complex network for preventing fault propagation in Web Service, we decided to construct a complex network module by using invocable relations between web services firstly. The invocable relation is determined by matching the input and output attributes of web services and the complex network model would be set up based on the invocable relations. Web services are taken as nodes, and nodes which satisfy invocable relation are linked together in the complex network model.

Definition 1 (Web Service) Web Service *S* is an atomic service and only has one operation. It is defined as a tuple *S<ID, Porttype, Operation, Message, QoS, Description>*, *ID* is the identification of *S*; *Porttype is* the protocol and data format definitions of special port type; *Operation<in, out>* is the description of operation provided by *S*, which includes a pair of input and output; *Message* is the type definition of communication data; *QoS represents the quality of S*; *Description* represents the functionalities realized by *S*.

Definition 2 (Invocable Relation) Assuming that Web Service S_1 has operation$_1$<in_1, out_1>, S_2 has operation$_2$<in_2, out_2>, if $S_1.\ out_1 \cap S_2.in_2 \neq \phi$,there is invocable relation between S_1 and S_2, and invocable degree between S_1 and S_2 is defined as:

$$IODegree = \frac{\left|S_1.out_1 \text{ I } S_2.in_2\right|}{\left|S_2.in_2\right|}.$$

Definition 3 (Web Service Complex Network) The model of Web Service Complex Network (*WSCN* for short) can be presented by a set V of nodes and a set E of edges, connected together as a graph denoted G = (V;E), each edge e∈E is the invocable relation connected to one pair of web services, one at each end.

Definition 4 (QoS on Edge) Assuming that S_1, S_2 in *WSCN*, there is invocable relation between S_1 and S_2, QoS of S_1 and S_2 are QoS_1 and QoS_2 respectively. The edge between S_1 and S_2 is $edge_{12}$, and its QoS is $\frac{QoS_1 + QoS_2}{2}$.

3 Fault Propagation on WSCN

In this section, we model the dynamic process of fault propagation in Web Service. Assuming that there are N services in WSCN, we define the Maximum-load Capacity and remaining capacity of Web Service S_i.

Definition 5 (Maximum-load Capacity of S_i) Let Maximum-load Capacity C_i of S_i be proportional to L_i (L_i is the degree of S_i): $C_i = \alpha L_i$, where α is the tolerance parameter.

Definition 6 (Remaining Capacity of S_i) Let Maximum-load Capacity of S_i be C_i , and the present load of S_i be N_i , the Remaining Capacity R_i of S_i satisfies: $R_i = C_i - N_i$.

When Web Service S_i fails to work, S_i would be removed from the network and its load would be redistributed over the rest of the network according to the invocable relations between services. The redistribution of load may cause the failure of other services, and lead to cascading failure finally. If the remaining services' load is less than their capacity, the cascade ends and the network return to a new balanced state. Let Web Service S_i be any node in WSCN, and its present load is L_i. Let set {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$} be Father Nodes of S_i, every service in set {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$} has connected with S_i, and the weight of these arcs are {$pweight_1$, $pweight_2$, ⋯⋯ , $pweight_m$} respectively. Let set {$child_1$, $child_2$, ⋯⋯ , $child_k$} be Child Nodes of S_i, S_i has connected with every service in set {$child_1$, $child_2$, ⋯⋯ , $child_k$}, and the weight of these arcs are {$cweight_1$, $cweight_2$, ⋯⋯ , $cweight_k$} respectively. In-degree and Out-degree of S_i are expressed as $IDeg_i$ and $ODeg_i$.

According to the composition rules in services, if one service fails to work, it needs to find others having similar functions to replace it. In WSCN, as the services connect with each other by invocable relations, one service may provide similar functions with children of its farther nodes or farther of its child nodes. Consequently, if S_i fails to work, its load would be distributed to child nodes of {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$} and father nodes of {$child_1$, $child_2$, ⋯⋯ , $child_k$} shown in figure 2. The load distributed to child nodes of {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$} is:

$$\Delta L_p = L_i \frac{IDeg_i}{IDeg_i + ODeg_i}.$$

The load distributed to father nodes of {$child_1$, $child_2$, ⋯⋯ , $child_k$} is:

$$\Delta L_c = L_i \frac{ODeg_i}{IDeg_i + ODeg_i}$$

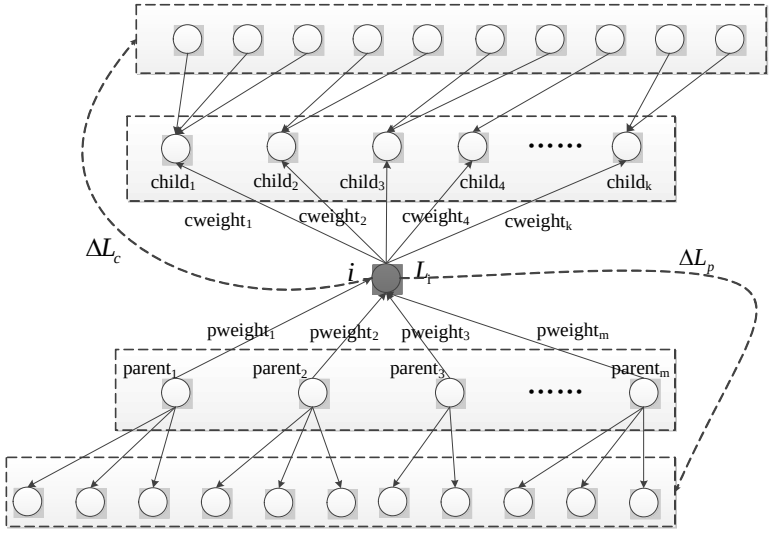


Figure 2 All of the nodes distributed loads of Service i.

As shown in figure 3, if $parent_j$ is one element in set {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$}, the load distributed to all of child nodes of $parent_j$ is:

$$\Delta L_{pj} = \Delta L_p \frac{pweight_j}{\sum_1^m pweight_j}$$

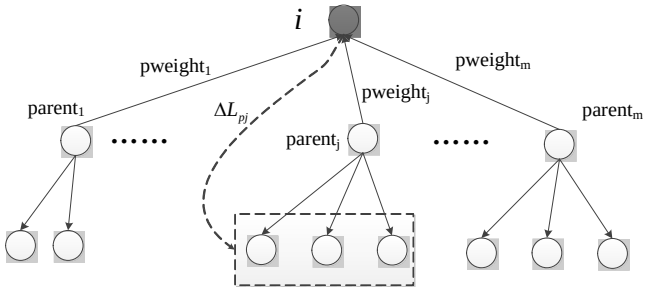


Figure 3 Some nodes distributed load of Service i

When load distributed to child nodes of {$parent_1$, $parent_2$, ⋯⋯ , $parent_m$} and father nodes of {$child_1$, $child_2$, ⋯⋯ , $child_k$}, we consider two different distribution strategies:

(1) Load distribution based on weight

If $parent_j$ is one element in set $\{parent_1, parent_2, \cdots\cdots, parent_m\}$, let $\{p_{j1}, p_{j1}, \cdots\cdots, p_{jn}\}(n \geqslant 1)$ be the child nodes of $parent_j$. The weight of arcs connected from every node in set $\{p_{j1}, p_{j1}, \cdots\cdots, p_{jn}\}$ to $parent_j$ are $pw_{j1}, pw_{j2}, \cdots\cdots, pw_{jn}$. If S_i fails to work, the load distributed to node $p_{jn'}$ $(1 \leqslant n' \leqslant n)$ is:

$$\Delta L_{jn'} = \Delta L_{pj} \frac{pw_{jn'}}{\sum_1^n pw_{jn'}}.$$

(2) Load distribution based on remaining capacity

If $parent_j$ is one element in set $\{parent_1, parent_2, \cdots\cdots, parent_m\}$, let $\{p_{j1}, p_{j1}, \cdots\cdots, p_{jn}\}(n \geqslant 1)$ be the child nodes of $parent_j$. The remaining capacity of every node in set $\{p_{j1}, p_{j1}, \cdots\cdots, p_{jn}\}$ to $parent_j$ are $pu_{j1}, pu_{j2}, \cdots\cdots, pu_{jn}$. If S_i fails to work, the load distributed to node $p_{jn'}$ $(1 \leqslant n' \leqslant n)$ is:

$$\Delta L_{jn'} = \Delta L_{pj} \frac{pu_{jn'}}{\sum_1^n pu_{jn'}}.$$

After load distribution, if node $p_{jk'}$ is over load, it would be removed from network and a new load distribution would happen. Other nodes in the network are dealt with the same strategies.

4 Experiments

In order to analyze the resistance of WSCN influenced by the size of network, different types of attacks and load allocation strategies, we select different numbers of Web Services to construct four WSCN in different sizes, and simulate the propagation process by analyzing their performance under small attack, large attack, random attack and calculated attack. We also use weight-based and spare-load-based load allocation methods of failed service to compare their influences on the whole network.

4.1 Experimental Data

According to the above definitions, *WSCN* is built by setting Web Service as node, and Invocable Relation between services as edge. Consequently, *WSCN* is a directed weighted network, and the direction is from invoking service to invoked service, and weight is computed by Invocable Degree. We choose the data set from Web Service Challenge 2009[21] as experiment data. In the process of constructing *WSCN*, firstly we parse files of wsdl and wsla and get services information, then computing the Invocable Degree between services, which could control the scale of the complex network model. We define $IODegree = 0.01$ when constructing the complex network module. We choose data set including 215, 426, 1157 and 2710 services to construct module, analyze their properties of small world by computing average path length and clustering coefficient, and scale-free properties by checking degree distribution. The results of computing average path length and clustering coefficient are shown in the following table.

Table 1 Attributes of Networks in Different Size

Network size	Average Path Length	Clustering Coefficient
215	4.357	0.298
426	3.546	0.103
1157	3.139	0.087
2710	3.004	0.088

According to the content in Table 1, all of the four modules have small average path length and big clustering coefficient, which shows that all of them satisfy the features of small world. In the nodes' degree distribution of four networks shown in Figure 1, most nodes' degree are small, while a small number of nodes' degree are big, and the degree distribution of four networks meet power-law distribution. It means that the four modules satisfy the feature of scale-free network. Consequently, the four different sizes of networks including 215, 426, 1157, and 2710 services are Complex Network.

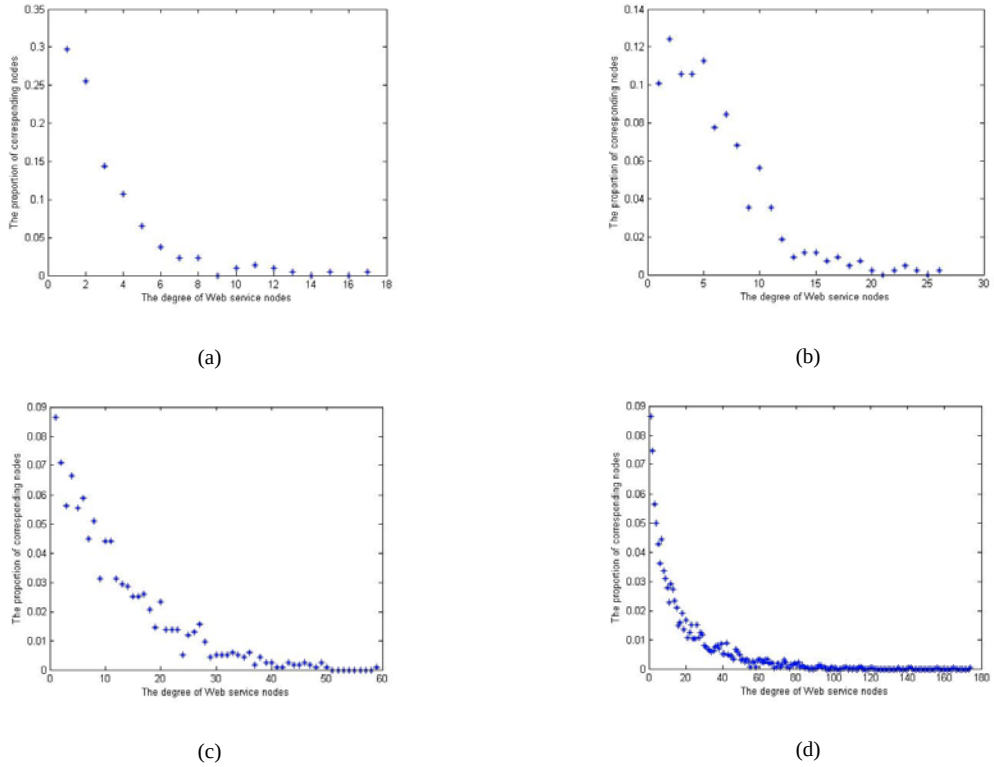


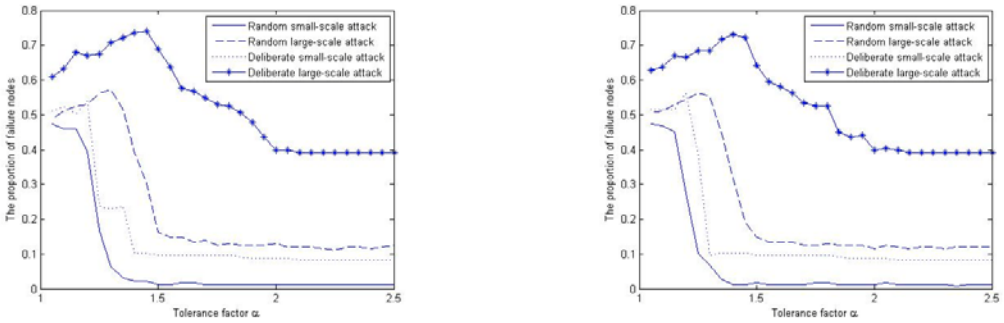
Figure 1 Nodes' degree distribution of four networks

We simulate fault propagation on four networks respectively, and the experiment results are analyzed from two aspects. One is the influence on the network of different attack types, and the other is the influence of load allocation strategy in different ways.

4.2 Influences on Four Networks of Different Attack Types

In order to analyze the influence on WSCN under different attack types, we do the experiment of different networks composed of 215, 426, 1157 and 2710 nodes under random small-scale attack, random large-scale attack, deliberate small-scale attack and deliberate large-scale attack. We want to find the relation between the proportion of failure nodes and the tolerance factor from experiments and the results of experiment is shown in Figure 4-7. In these figures, the horizontal axis represents the tolerance factor, and the vertical coordinate represents the proportion of failure nodes. In every figure, there are four curves representing the relation between proportion of failure nodes and tolerance factor under four kinds of attack respectively. We would analyze the relations between proportion of failure nodes and tolerance factor the influence on different size of networks under different types of attacks in the following paper.

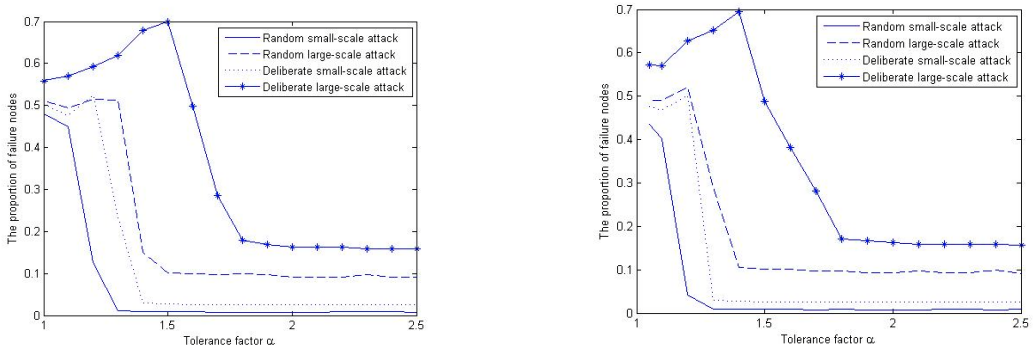
For the network of 215 nodes, as shown in Figure 4(a), when the tolerance factor is 1.5, the proportion of failure nodes is more than 60% in the deliberate large-scale attack and less than 20% in the large-scale random attack. It shows that the network can overcome the random attack more than the deliberate one. With the deliberate large-scale attack, whether weight-based or remaining-space-based distribution is chosen, the proportion of failure nodes is more than 40%. It implies that the deliberate large-scale attack has a huge influence on the network.



(a) Weight-based Distribution (b) Remaining space-based Distribution

Figure 4: Proportion of failure nodes as a function of tolerance factor, for a 215-node network and four attack types

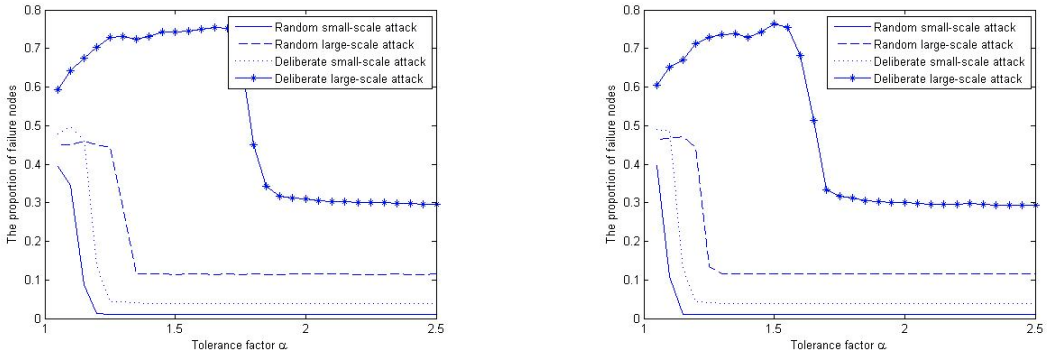
For the network of 426 nodes, as shown in Figure 5, the comparison of random small-scale or large-scale attack curves shows that large-scale attacks have more effect on the network than small-scale attacks. As shown in Figure 5(a), if the proportion of failure nodes is desired to be less than 10%, the tolerance factor needs to be at least 1.2 in the random small-scale attack, but the factor must be increased to 1.4 in the large-scale random attack. The comparison of the deliberate small-scale and large-scale curves shows that for tolerance factor above 1.4, there will be fewer than 5% failing nodes under the deliberate small-scale attack. With deliberate large-scale attack, if the failing nodes in the network are desired at fewer than 20%, the tolerance factor must be chosen above 1.8. From the two curves shown in Figure 5(a) under the deliberate attack, it can be concluded that with the increasing of the tolerance factor, the proportion of failure nodes firstly increases, and then decreases after reaching a critical value. A similar result is seen in Figure 5(b) when considering the remaining space-based distribution.



(a) Weight-based Distribution (b) Remaining space-based Distribution

Figure 5: Proportion of failure nodes as a function of tolerance factor, for a 426-node network and four attack types.

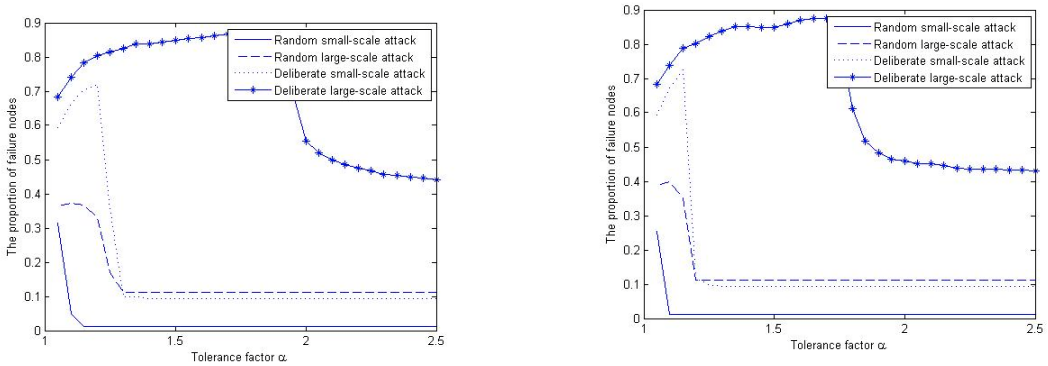
In the network of 1157 nodes, as shown in Figure 6(a), with the random small-scale attack, the proportion of failure nodes is 40% for a tolerance factor of 1.0. The proportion of failure nodes reduces from 40% to 1% as the tolerance factor is increased from 1.0 to 1.3. If the tolerance factor is bigger than 1.3, there are almost no failure nodes in the network.



(a) Weight-based Distribution (b) Remaining space-based Distribution

Figure 6: Proportion of failure nodes as a function of tolerance factor, for a 1157-node network and four attack types.

For the network of 2710 nodes, as shown in Figure 7(b), if the deliberate large-scale attack happens, the proportion of failure nodes increases gradually with the increasing of tolerance factor from 1.0 to 1.7. When the tolerance factor is 1.0, the proportion of failure nodes is 70%. When the tolerance factor is 1.7, the proportion of failure nodes is nearly 90%. But the proportion of failure nodes reduces gradually with the increasing of the tolerance factor from 1.7 to 2.0. When the tolerance factor reaches 2.0, the proportion of failure nodes asymptotically approaches 40% and does not decrease substantially with further increase of the tolerance factor. Therefore, the whole network would break down with the deliberate large-scale attack.



(a) Weight-based Distribution

(b) Remaining space-based Distribution

Figure 7: Proportion of failure nodes as a function of tolerance factor, for a 2710-node network and four attack types.

From the above comparisons of different attack type in networks having different size, we could obtain the following conclusions.

Firstly, the network is affected differently by different types of attacks, and resists random attacks more than deliberate ones. If deliberate attacks happen in the network, the propagation area is wide. If the tolerance factor is close to 1, almost all of the nodes in the network would fail.

Secondly, the number of failure nodes has an effect on the network's behavior. If the number of initial failure nodes is small, there is only a small effect on the network. Under the same attack conditions, if more nodes fail initially, the network would face a large area of collapse.

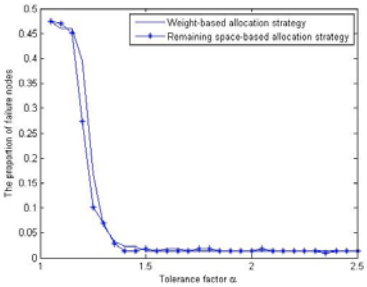
Thirdly, there is critical range [a,b] of tolerance factor for different sizes of network and different types of attacks. If the tolerance factor is smaller than a, one finds the whole network will collapse. If the tolerance factor is bigger than a, and smaller than b, the resistance of the network to failure improves as the tolerance factor increases. If the tolerance factor is larger than b, the number of node failures does not further decrease substantially.

4.3. Influences of Load Allocation Strategy in Different Ways

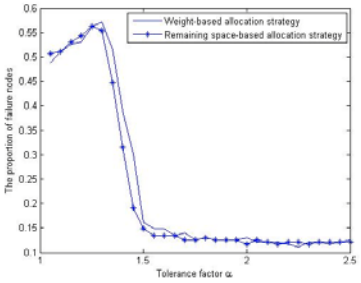
In order to analyze the influence on the network in different load allocation strategy, the networks of 215, 426, 1157 and 2710 nodes are researched under the weight-based allocation strategy and the remaining space-based one. The relation between the proportion of failure nodes and the tolerance factor is calculated. In Figure 8-11, horizontal axis gives the tolerance factor, and the vertical axis shows the resulting proportion of failure nodes. The two curves in the figure represent the relation between the proportion of failure nodes and the tolerance factor under the weight-based allocation strategy and the remaining space-based one respectively.

For the network of 215 nodes, as shown in Figure 8, for different load allocation strategy curves under the random small-scale attack, the random large-scale one, the deliberate small-scale one and the deliberate large-scale one, the curves of the weight-based and the remaining space-based allocation

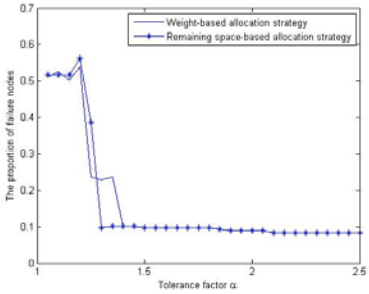
strategies overlap. There is no obvious difference of failure proportion between the two allocation strategies for this 215-node network.



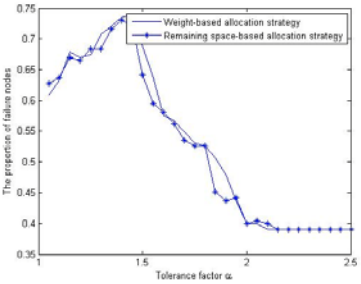
(a) Random Small-Scale Attack



(b) Random Large-Scale Attack



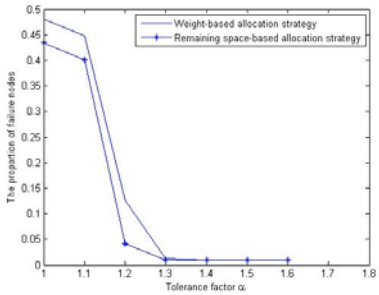
(c) Deliberate Small-Scale Attack



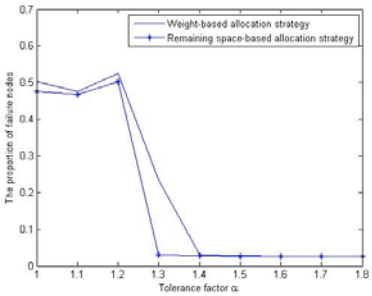
(d) Deliberate Large-Scale Attack

Figure 8 Comparison of the proportion of failing nodes for different load allocation strategies on a network of 215 nodes

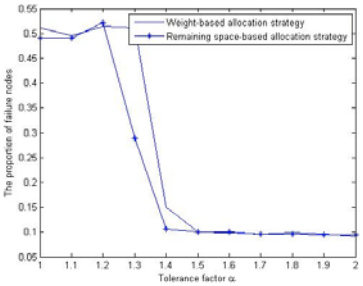
In the network of 426 nodes shown in Figure 9, the comparison of curves based on two strategies shows that the network can usually overcome more failure under the remaining space based allocation strategy. For the deliberate small-scale attack in Figure 9(c), when the tolerance factor is 1.3, the weight-based allocation strategy will result in roughly 20% of nodes failing, but the remaining space-based strategy is about 10 times better. However, when considering all curves in figure 9, the network of 426 nodes shows only minor differences between the two different load allocation strategies.



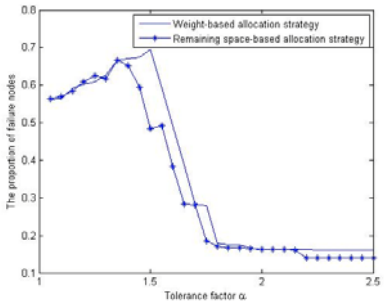
(a) Random Small-Scale Attack



(c) Deliberate Small-Scale Attack



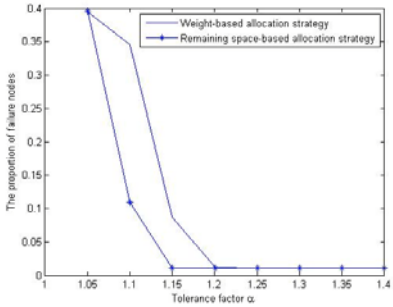
(b) Random Large-Scale Attack



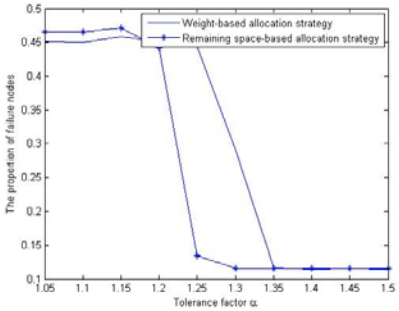
(d) Deliberate Large-Scale Attack

Figure 9 Comparison of the proportion of failing nodes for different load allocation strategies on a network of 426 nodes

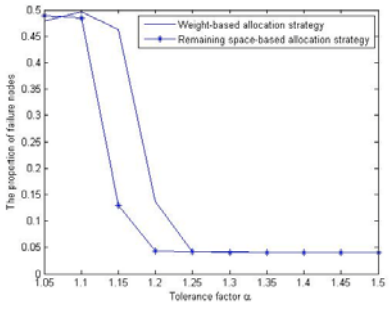
For the network of 1157 nodes shown in Figure 10(d), when the deliberate large-scale attack happens, if the tolerance factor is 1.7, the proportion of failure nodes is 30% with the remaining space based allocation strategy. But with the weight-based one, the proportion is 70%. It shows that the remaining space-based allocation strategy can improve the robustness of the network for the failure.



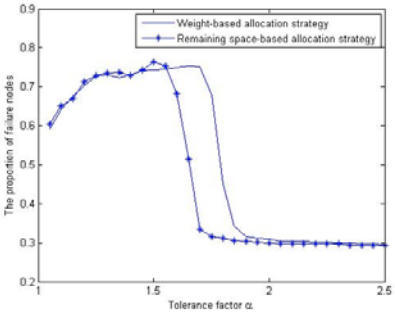
(a) Random Small-Scale Attack



(b) Random Large-Scale Attack



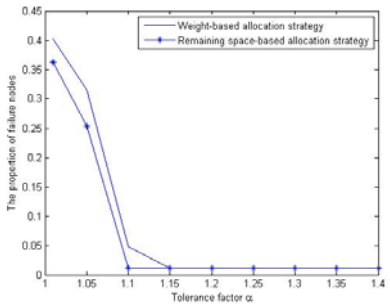
(c) Deliberate Small-Scale Attack



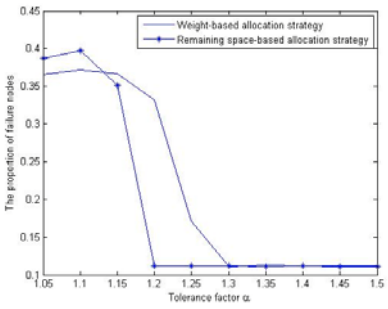
(d) Deliberate Large-Scale Attack

Figure 10 Comparison of different load allocation strategies on the network of 1157 nodes

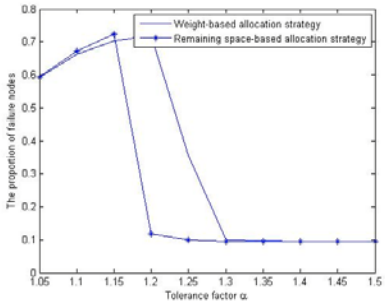
For the network of 2710 nodes shown in Figure 11, when the random small-scale attack happens as shown in Figure 10(a), there is no obvious difference between two load allocation strategies. When the deliberate large-scale attack happens as shown in Figure (d), there is much difference between the curves based on the weight and remaining space. When the tolerance factor is 1.85, the remaining space-based allocation strategy will lead to about 45% of failure nodes, but the weight-based one will lead to about 85%.



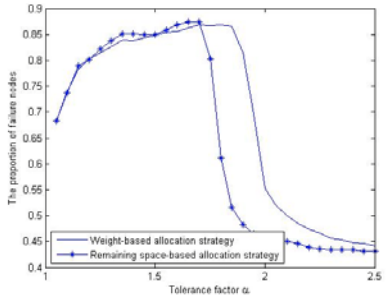
(a) Random Small-Scale Attack



(b) Random Large-Scale Attack



(c) Deliberate Small-Scale Attack



(d) Deliberate Large-Scale Attack

Figure 11 Comparison of different load allocation strategies on the network of 1157 nodes

From the above comparisons of different attack types in networks having different sizes, we can obtain the following conclusions.

Firstly, when the failure happens in WSCN, the weight based and remaining space-based allocation strategies have different affects on the network. The network resists the failure more strongly when adopting the remaining space-based one.

Secondly, if the network size is small, there are no clear differences between the two allocation strategies. For networks of 215 or 426 nodes, the dependence of failures on tolerance factor is essentially the same under all different types of attacks. The reason for this phenomenon is that there is very little difference between the maximum and minimum value of node's degree. The node's initial load and space is associated with its degree, and the little difference of degree leads to little difference of remaining space. If the size of the network becomes lager, there would be larger difference between

maximum and minimum value of node's degree, and the network's robustness is much stronger when incorporating the remaining space-based strategy.

5 Discussions

The results of the experiments had shown that the network has better resistance for small scale failure than big scale one, and resisted stronger for random attack than deliberate one. If the service got failed, the remaining space based load allocation strategy on it had higher robustness than weight based one. From the simulation of the fault propagation model on WSCN, we'd better control the number of failed services at the same time, avoided the happening of deliberate attack on them, and choose the remaining space based load allocation strategy for the failed service node in the real environment.

6 Conclusions and Future Work

In this paper, we have proposed a new approach to prevent fault propagation from the view of the whole network. It put forwards a Web Service Complex Network model based on the composition relations, presented the dynamic process of its fault propagation, and analyzed its resistance influenced by different sizes of network, different types of attacks and load allocation strategies by experiments. The simulation experiment had been done to analyze the robustness of large-scale attacks, small-scale attacks, random attack and deliberate attack to the model and compare the influences of the weight-based and remaining-spare-based load allocation strategy to the failed service nodes. The simulation of fault propagation for Web Service could set example for preventing and reducing probabilities of collapse in the service network.

As the future work, we will choose open services provided in our university as the nodes in the Web Service Complex Network, for example, services from teaching management, personnel management, research project management and finance management systems. We would produce a large amount of services based on above services by changing parameters of services' interface, and deploy them on different servers. We would construct a real Web Service Complex Network based on these services and implement our approach on the network to conform the effectiveness of our approach.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61402090, 61374178, 61202085, 61100027), the Liaoning Provincial Natural Science Foundation of China (Grant No. 201202076), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No.20120042120010), the Ph.D. Start-up Foundation of Liaoning Province, China (Grant Nos. 20121001, 20121002), and the fundamental research funds for the central university (Grant No. N120817002).

References

1. Papazoglou, M. P. (2003, December). Service-oriented computing: Concepts, characteristics and directions. In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on (pp. 3-12). IEEE.

2. Erl, T. (2004). Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR.
3. Hwang, S. Y., Lim, E. P., Lee, C. H., & Chen, C. H. (2008). Dynamic web service selection for reliable web service composition. Services Computing, IEEE Transactions on, 1(2), 104-116.
4. Chan, K. M., Bishop, J., Steyn, J., Baresi, L., & Guinea, S. (2009, January). A fault taxonomy for web service composition. In Service-oriented computing-ICSOC 2007 Workshops (pp. 363-375). Springer Berlin Heidelberg.
5. Y. Yan, P. Dague, Y. Pencole, and M.O. Cordier. A model-based approach for diagnosing fault in web service processes. International Journal of Web Services Research (IJWSR), 6(1):87-110, 2009.
6. A. Erradi, P. Maheshwari, and V. Tosic. Recovery policies for enhancing web services reliability. In Web Services, 2006. ICWS'06. International Conference on, pages 189-196. IEEE, 2006.
7. W. He. Recovery in web service applications. In e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on, pages 25-28. IEEE, 2004.
8. Z. Zhu, J. Li, Y. Zhao, and Z. Li. Scenetester: A testing framework to support fault diagnosis for web service composition. In Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on, pages 109-114. IEEE, 2011.
9. G. Friedrich, M. Fugini, E. Mussi, B. Pernici, and G. Tagni. Exception handling for repair in service-based processes. Software Engineering, IEEE Transactions on, 36(2):198-215, 2010.
10. Motter, A. E., & Lai, Y. C. (2002). Cascade-based attacks on complex networks. Physical Review E, 66(6), 065102.
11. Crucitti, P., Latora, V., Marchiori, M., & Rapisarda, A. (2004). Error and attack tolerance of complex networks. Physica A: Statistical Mechanics and its Applications, 340(1), 388-394.
12. Yau, S. S., Ye, N., Sarjoughian, H. S., Huang, D., Roontiva, A., Baydogan, M., & Muqsith, M. A. (2009). Toward development of adaptive service-based software systems. Services Computing, IEEE Transactions on, 2(3), 247-260.
13. J. Zhou, K. Cooper, I. Yen and R. Paul, Rule-Base Technique for Component Adaptation to Support QoS-Based Reconfiguration, Proc. Ninth IEEE Int',l Symp. Object-Oriented Real-Time Distributed Computing, pp. 426-433, May 2005.
14. Bak, P., Tang, C., & Wiesenfeld, K. (1987). Self-organized criticality: An explanation of 1/f noise. Physical Review Letters, 59(4), 381-384.
15. Motter, A. E. (2004). Cascade control and defense in complex networks. Physical Review Letters, 93(9), 098701.
16. Moreno, Y., Gómez, J. B., & Pacheco, A. F. (2002). Instability of scale-free networks under node-breaking avalanches. EPL (Europhysics Letters), 58(4), 630.
17. Kinney, R., Crucitti, P., Albert, R., & Latora, V. (2005). Modeling cascading failures in the North American power grid. The European Physical Journal B-Condensed Matter and Complex Systems, 46(1), 101-107.
18. Bao, Z. J., Cao, Y. J., Ding, L. J., Han, Z. X., & Wang, G. Z. (2008). Dynamics of load entropy during cascading failure propagation in scale-free networks. Physics Letters A, 372(36), 5778-5782.
19. Crucitti, P., Latora, V., & Marchiori, M. (2004). A topological analysis of the Italian electric power grid. Physica A: Statistical Mechanics and its Applications, 338(1), 92-97.
20. Liu, A., Li, Q., Huang, L., & Xiao, M. (2010). Facts: A framework for fault-tolerant composition of transactional web services.Services Computing, IEEE Transactions on,3(1), 46-59.
21. S. Kona, A. Bansal, M. B. Blake, S. Bleul, T. Weise, WSC-2009: A Quality of Service-Oriented Web Services Challenge, IEEE Computer Society, pp.487-490, Vienna, Austria, 2009.