
The Optimal Resource Self-configuration Method of Cognitive Network for Survivability Enhancement

Jian Wang^{1,*}, Guosheng Zhao², Zhongnan Zhao¹ and Zhixin Li¹

¹*School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China*

²*School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China*

E-mail: wangjianlydia@163.com

**Corresponding Author*

Received 14 November 2019; Accepted 19 April 2020;

Publication 29 July 2020

Abstract

In view of that general lack of intelligence and flexibility of the existing network resource allocation methods in the case of time-varying environments and diversified requirements, an efficient self-configuration method is put forward to optimize the allocation of resources and improve the survivability of system. First of all, the utility function of consumption domain is introduced as an indicator to pre-arrange the priority of user's QoS, as a result, the utility maximization of the system under resource constraint is obtained. Then, based on this definition, a multidimensional dynamic programming framework is proposed to define and describe the self-configuration process, and the problem model is constructed under certain constraints. Furthermore, the adaptive adjustment and configuration of resources are implemented by determining the priority sequence of user services, finding the optimal resource configuration scheme, and optimizing the time configuration window. Finally, The simulation results show that the

Journal of Web Engineering, Vol. 19_3-4, 503–520.

doi: 10.13052/jwe1540-9589.19346

© 2020 River Publishers

proposed method is superior to the traditional resource allocation scheme in terms of system reliability, connectivity, broadband utilization, average response time and transmission rate, which improves the system's ability to adapt to the environment intelligently and survivability effectively.

Keywords: Survivability enhancement, cognitive network, self-configuration, utility function.

1 Introduction

Survivability [1] refers to the ability of a system to perform its mission-critical tasks in a timely manner when it is subject to attacks, failures, and accidents. The idea of survivability can be summarized as adaptive "capacity", that is, the system can be invaded, some components can be damaged, and even some components are not completely reliable, but as long as the system can reasonably allocate resources in structure and reorganize resources under attacks, with self-protection capabilities such as self-optimization, self-maintenance, self-regulation and functional semantic redundancy, it can still ensure the completion of key services timely and repair the damaged key services. In the case of hardware failures, software bugs, human error or even malicious attacks, how to enhance the survivability of all kinds of mission-critical systems effectively, so that they can run according to the expected state in different environments, has become a key problem to be resolved.

Survivability enhancement is an enhanced design and proposed countermeasures for the weak links of the system, such as intrusion resistance, identification and recovery. The existing system survivability enhancement research mainly has the following problems: limited to defensive and bastion technologies and methods (such as redundancy technology, Byzantine threshold mode, key and isolation technology, etc.). These are static and passive survivability enhancements, which are too clumsy or even helpless to deal with the events that may lead to the unavailability of system services, and it is difficult to meet the needs of practical applications. Therefore, an automatic flexible and fine-grained active management method is expected to enhance the system survivability. Cognitive Computing is a new way to solve the self-management of complex heterogeneous systems, which has received high attention from the military and prominent international companies since it was proposed. Its goal is to make the system have the ability of self-awareness, reasoning, interaction and learning, etc. By introducing the cognitive mechanism into the study of survivability enhancement,

it will make qualitative changes in the concept and idea of survivability research and design. The cognitive network has the ability to sense the current network state and the external environment, and utilizes the perceived network state and external environment information to correctly plan, decide and execute the configurable elements of the network, so as to achieve the goal of “using technology to manage technology” of the network system [2–4]. Self-configuration [5, 6] based on external environment information and system behavior of aware computing, study how to make the system analyze and infer the trend of environmental changes, the distribution trend of system resources, and the prior knowledge of users in the process of running, respond to dynamic changes in its external environmental information and user application requirements, and make appropriate configuration response according to survivability requirements. Thus, the system survivability in the uncertain environment can be effectively enhanced.

Due to its good characteristics in self-management and evolution ability, the cognitive network has received extensive attention from governments and research institutions, including IBM, HP, Sun, and other commercial companies. So far, research on the application of autonomic computing in the field of networks has obtained certain achievements. In literature [7], a dynamic QoS self-configuration framework was proposed to solve the problem that user’s QoS can not be directly applied to next-generation networks (NGN) such as integrated services and differentiated services in the traditional network system. This framework incorporated utility function and the interrupt mechanism, where the utility function was used to indicate the priority of the user’s QoS, and the interrupt mechanism was used to dynamically correct the priority in the process of transmitting the data packet. Because of the self-configuring active feature, the problem of user QoS degradation in the case of network congestion had also been improved. A highly flexible application component structure was proposed in literature [8], which was mainly applied to the control application system. It was an architecture that can be automatically deployed in real time. The architecture was highly scalable, by embedding multiple different dynamic decision points simultaneously in the appropriate parts of the software, and it took appropriate configuration actions to automatically solve the problems caused by dynamic decision-making, and can evaluate the current software running status. A trust-tolerant and scalable self-configuration algorithm for organic computing systems was proposed by Nizar et al. [9]. The purpose was to use the service load on the average distribution node as a typical load balancing scheme. On the other hand, adding services with different levels of importance to the nodes

made it easier for more critical services to be assigned to nodes with high reliability. At the same time, the algorithm also included a fault handling mechanism that enabled the system to continue to host the service even in the presence of a fault. A self-configuration scheme was proposed by Yaqub et al. [10] to select the optimal wavelet decomposition level, adaptively select the training set and features, and optimize the wavelet decomposition level selection to maximize the energy bandwidth of the fault signature signal. The support vector machine was used to construct the fault diagnosis model, and its performance and robustness were verified for data of different severity. Literature [11] summarized the architecture of self-organizing networks, and discussed the issue of autonomous physical unit identifier allocation and radio resource allocation schemes and self-organizing network optimization strategies. An automatic network deployment mechanism based on SDN was proposed by Mohamed et al. [12]. By discovering the reachable SDN controller and then choosing to connect to the most appropriate one, the new SDN was added and set in the wireless router when the controller was unavailable. An autonomous configuration framework for middleware was proposed in literature [13], in which the system automatically configured and adjusted the configuration parameters of the system through the perceived dynamic changes of the load. The framework mainly seek a method to optimize resource allocation through a performance change prediction model based on a hierarchical queuing network, so that the system had self-adjusting ability and can adapt to environmental changes to meet user QoS. In literature [14], the mobile agent technology and P2P were combined, and the self-configuration feature was added. Based on this, a self-configuration system architecture based on policy was proposed, and a BestPeer P2P system was established. The system can independently configure various components and components according to a predetermined strategy, and at the same time a self-configuring system evaluation method was proposed, which can comprehensively analyze and evaluate the system according to various automatic configuration evolution examples of the software system. In literature [15], an adaptive configuration framework of database system was put forward to address the problem that the database system load with high dynamic changes in the cloud computing environment can not guarantee the quality of service. The framework can automatically adjust the system resource configuration according to the dynamic changes of the load, thus ensuring the quality of service required by users.

As mentioned above, the existing configuration methods generally lack of intelligence and flexibility. Firstly, once the resource allocation method is

determined for the system, the system can no longer make adaptive changes, and then can only rely on manual intervention by users or administrators. Secondly, the execution status and results of the scheduling policy cannot be fed back to the service sender in time, and adjustments cannot be made in time according to various conditions that occur. In view of the above problems, in this paper, an efficient resource self-configuration method is studied to optimize the allocation of resources. Resources are dynamically allocated and divided according to external environment changes and different service requirements of users. Meanwhile, the core issues such as dynamic expansion and optimization of self-configuration behavior, consistency and integrity of self-configuration are considered. With the main goal of diversified service demand and efficient use of system resources, the services provided by the system are maintained at a satisfactory level without human intervention, meeting the requirements of system survivability growth.

2 Problem Definitions

2.1 Introduction of Utility Functions

The emergence of utility theory promotes the development of consumption theory. Utility represents a kind of subjective feeling, which is an abstract concept. It is impossible for users to get the specific quantity size through measurement, but they can only use the comparison method and judge the order of superior by the given ordinal number. It is often used to analyze the satisfaction of consumers in the whole process of consumption. Consumers can sort the multiple feasible consumption schemes into a sequence similar to the real number size. And in this process, the utility function can describe the superior judgment of the consumption scheme.

At present, the application of utility function in the field of computer has become a hot topic, and many scholars have paid attention to it. In the network field, the utility function can be used to measure the satisfaction of the users who use the system. The order of user QoS priorities is pre-arranged according to the utility function, so that the utility of the system under the network condition is maximized under the constraints of resources. In the process of studying the configuration problem of network resources, the QoS grading of user services and the efficiency of network information systems are both key indicators for evaluating the performance of network resource configuration. If only considering the QoS maximization of the user service or the high efficiency of the network system, any single target will

cause problems in the solution result. The utility function is used to describe the user's QoS in this paper, which can more accurately describe the current system state, and in turn, the strategy and guidance are made for the efficient operation of the system in this state.

2.2 Description of Self-configuring Process

A self-configuring system mainly includes five aspects: external environment factors, managed resources, environment awareness, self-configuration managers, and self-configuration strategies. Among them, the managed resources belong to the passive allocation, and the self-configuration manager and the self-configuration strategy belong to the main initiator and performer of the configuration activity. On the basis, the environment awareness is the bridge between them. The main task of environment awareness is to be responsible for forwarding the underlying environmental information to the upper layer application and providing configuration motivation. The self-configuration manager is responsible for analyzing, planning, and deciding the collected environmental information, and finally, the self-configuration behavior is completed dynamically according to the self-configuration policy. Although there are many ways to implement self-configuration technology, they are all related to specific areas. Traditional self-configuration techniques have been unable to adapt to the current complex and changeable computing environment.

In order to effectively overcome the difficulties in the process of resource allocation, a multi-dimensional dynamic programming framework is constructed in this paper, which solves the problem through modeling analysis under certain constraints and constraints. Thus, three sub-problems to be solved are obtained: determining the priority sequence of user services, finding the optimal resource allocation scheme, optimizing the time allocation window, and then solving them one by one.

The resource allocation problem is described as follows: Suppose there is a service or application set $Services = (S_1(t), S_2(t), \dots, S_{|S|}(t))$, where $|S|$ represents the number of services or applications, and $S(t)$ means that the service set assigns the time attribute to each service or application. The resource set in the network is represented by $Resource = (R_1, R_2, \dots, R_{|R|})$, and the size of $|R|$ represents the number of different resources. Q_s indicates the profit and loss rate if service S is not implemented or fails. Similarly, if each type of resource can be effectively allocated to a service or application that needs it, it will produce corresponding benefits inevitably, namely

efficiency. P_{RS} represents the efficiency of resource R for service S , where $p_{RS}(t) \in [0, 1]$. When P_{RS} is 0, it means that the resource has no meaning for the service, and does not have the allocability. On the contrary, P_{RS} is not 0, indicating that the resource is not 0 for the completion efficiency of the service, and is assignable. $SA(R)$ represents the set of services and applications for which resource R are assignable, $RE(S)$ represents a set of different kinds of resources that the service or application S needs and can be assigned. On the issue of resource allocation, the decision variable is set to be $x_{RS}(t)$, indicating the number of resource R allocated to the service S at time t . Among them, if $x_{RS}(t) = 0$, it means that no resource allocation operation is performed. In the description of time, T_S^S represents the earliest arrival time of the request of S , T_S^E represents the latest accepted time of the request; $TS_S = [T_S^S, T_S^E]$ represents the effective start time of service S ; T_{SR}^S represents the earliest point when resource R can provide service for S , and T_{SR}^E correspondingly represents the latest time point; The time period $TR_{RS} = [T_{RS}^S, T_{RS}^E]$ indicates the effective time when resource R can successfully provide resources to service S . In addition, the total number of resource R is represented by U_R , and the actual number available at time t is represented by $U_R(t)$.

In the configuration process of network resources, the following challenges are faced: (1) Resource constraints. When allocating various resources at time t , it is necessary to ensure that the requested resources do not exceed the actual available amount of the current resources; (2) Time constraints. Any resource must be allocated to the various services in a valid allocation time window; (3) Strategy constraints. That is to say, ensure that all allocation schemes are valid. It can be expressed by formula (1):

$$\left\{ \begin{array}{l} \sum_{S(t) \in Services} x_{RS}(t) \leq U_R(t), \forall R \in Resource \\ t \in [T_{RS}^S, T_{RS}^E], \forall x_{RS}(t) \geq 1 \\ [T_{RS}^S, T_{RS}^E] \subseteq [T_S^S, T_S^E] \end{array} \right. \quad (1)$$

The objective function is expressed by Equation (2) as follows:

$$f = \max \left\{ \prod_{S(t) \in Services} (1 - Q_S \prod_{R \in RE(S)} (1 - p_{RS})^{x_{RS}(t)}) \right\} \quad (2)$$

The model aims to maximize the number of completed services and achieve the highest revenue. While taking into account the minimization of

the profit and loss rate, the resource constraints, strategy constraints and time constraint are taken as constraints of the model to find optimal allocation of resources.

3 Self-configuration Algorithm of Cognitive Network

3.1 Determine the Priority Sequence of User Services

The determination of service priority sequence is to reasonably arrange the order in which resources required by the service and application are allocated. In the research process of this paper, the utility function index is used to calculate the accepted priority of service and application, and then sorted according to the size of the indicator, that is, the service with higher priority is allocated resources first.

The utility function of user u is defined as UI_u , which is expressed by formula (3):

$$UI_u = U_u(SE_u, DE_u, T_u, \partial_u) \quad (3)$$

where, SE_u represents the service horizontal space, DE_u represents the demand horizontal space, so the type of u is represented by SE_u and DE_u , that is to say, in this network environment, SE_u and DE_u constitute a “many-to-many” mapping relationship; T_u represents the time factor, and is evaluated from the perspective of time; ∂_u represents the correction factor, and the system appropriately corrects the user service priority value $Priority$, that is, the QoS value, according to different network environments. It can be seen that the optimization goal of the system can be expressed by the following formula (4):

$$UI_{\max} = \max_u \sum_u U_u(SE_u, DE_u, T_u, \partial_u) \quad (4)$$

$UI(R)$ is used to represent the utility function of resource R , and R_u^* represents the resource allocation scheme when the utility is the largest, expressed by the following formula (5):

$$\begin{cases} \sum_u UI(R) = \sum_u U_u(SE_u, DE_u, T_u, \partial_u) \\ R_u^* = \arg \max_R \sum_u UI(R_u) \\ s.t. \sum_u R_u \leq U_R(t) \end{cases} \quad (5)$$

However, this function for evaluating the priority does not take into account the network environment of system and the status of network transmission, so it cannot be used as the final decision function, and the priority sequence of services needs to be corrected. To ensure that the delay can be within the effective range, the correction factor ∂_u in *Priority* is described by the obstacle function, which is expressed by the following formula (6):

$$\partial_u = \frac{[d_{\max} - d_u(t)]^{1-\sigma_u}}{1 - \sigma_u} \quad (6)$$

where, d_{\max} is the maximum value of the delay acceptable to the user service, $d_u(t)$ is the actual delay of the current time t , and σ_u is the penalty factor that can be adjusted. A higher value of σ_u means that the system will give a higher compensation for the immediate timeout delay, and the priority of the user service will accordingly be higher. Therefore, the corrected priority of the user service can be expressed by Equation (7):

$$PRI_u = [Priority + \partial_u] \quad (7)$$

among them, *Priority* represents the service priority obtained by sorting according to the utility function without considering the current network environment.

3.2 Finding the Optimal Resource Allocation Scheme

In this paper, the self-configuration problem of resources in the network environment is mainly studied, which is divided into multiple stages of solving process, and dynamic programming is used to solve the problem. In order to improve the efficiency of decision-making, the allocation process of multiple types of resources can be divided into multiple stages, that is, a continuous decision variable is transformed into a number of sub-phases, each of which is responsible for allocating resources to applications or services. Compared with the general nonlinear programming method, using dynamic programming to solve the problem of resource allocation will greatly reduce the time complexity and improve the efficiency of the solution.

Since there are $|R|$ kinds of resources to be allocated, the general one-dimensional dynamic programming can not solve the resource allocation problem well, so the multi-dimensional dynamic programming method is adopted. At this time, the decision variables are converted into vectors, and the decision sets are transformed into matrices accordingly. And the

decision of each stage in the method appears in a multi-dimensional form, and its dimension is equal to $|R|$, namely the number of kinds of resources. The phase variable ω in each sub-phase is divided according to the service sequence that must be satisfied in each phase. $\omega = 0, 1, \dots, |\omega|$, the ω th phase corresponds to the resource allocation process of the ω th service or application.

The state variable is denoted as $(ST_{1S}, ST_{2S}, \dots, ST_{|R|S})$, where each component ST_{RS} represents the number of resource R remaining after the allocation of the first to S -1th service, that is, the number of resource R to be allocated to the S th to the $|S|$ th applications in the future, where $R = 1, 2, \dots, |R|$, $S = 1, 2, \dots, |S|$. The decision variables are described by $(x_{1S}, x_{2S}, \dots, x_{|R|S})$, each component x_{RS} represents the total amount of resource R allocated to the S th service or application in the S th to $S + \theta$ th stages, and θ represents the maximum number of phases occupied by resource R allocated to the ω th service request, where $R = 1, 2, \dots, |R|$, $S = 1, 2, \dots, |S|$.

The state transition relationship can be expressed by Equation (8):

$$\tilde{x}_R = ST_{RS} - x_{RS} \quad (8)$$

The above indicates the number of resource R allocated to the $S + 1$ th to the $|S|$ th services.

Through the above analysis, a dynamic programming model is obtained, as shown in Equations (9) and (10):

$$f_{|S|}(ST_{1|S|}, ST_{2|S|}, \dots, ST_{|R||S|}) = 1 - RE_{|S|} \prod_{R=1}^{|R|} (1 - p_{R|S|})^{x_{R|S|}} \quad (9)$$

$$\begin{aligned} & f_{\omega}(ST_{1\omega}, ST_{2\omega}, \dots, ST_{|R|\omega}) \\ &= \max_{\substack{0 \leq x_{1\omega} \leq ST_{1\omega} \\ \dots \\ 0 \leq x_{|R|\omega} \leq ST_{|R|\omega}}} \left\{ \left[1 - RE_{\omega} \prod_{R=1}^{|R|} (1 - p_{R\omega})^{x_{R\omega}} \right] f_{\omega+1} \right. \\ & \quad \left. (ST_{1\omega} - x_{1\omega}, ST_{2\omega} - x_{2\omega}, \dots, ST_{|R|\omega} - x_{|R|\omega}) \right\} \quad (10) \end{aligned}$$

In order to improve the computational efficiency of the solution, the approximate dynamic programming method is used to optimize the resource

allocation, and the corresponding mathematical model can be obtained by using Equations (11) and (12) as follows:

$$f_{|S|}(ST_{1|S|}, ST_{1|S|}, \dots, ST_{|R||S|}) = Q_{|S|} \prod_{R=1}^{|R|} (p_{R|S|})^{x_{R|S|}} \quad (11)$$

$$\begin{aligned} & f_{\omega}(ST_{1\omega}, ST_{2\omega}, \dots, ST_{|R|\omega}) \\ &= \max_{\substack{0 \leq x_{1\omega} \leq ST_{1\omega} \\ \dots \\ 0 \leq x_{|R|\omega} \leq ST_{|R|\omega}}} \left\{ \left[Q_{\omega} \prod_{R=1}^{|R|} (p_{R\omega})^{x_{R\omega}} \right] f_{\omega+1} \right. \\ & \left. (ST_{1\omega} - x_{1\omega}, ST_{2\omega} - x_{2\omega}, \dots, ST_{|R|\omega} - x_{|R|\omega}) \right\} \quad (12) \end{aligned}$$

3.3 Optimizing the Configuration Time Window

Assume that the priority sequence $SE = (S_1, S_2, \dots, S_x, \dots, S_N)$ of the service or application is now available. The resource allocation scheme for each service is $F = (F_1, F_2, \dots, F_x, \dots, F_N)$, and the resource allocation time for each service is $T = (T_1, T_2, \dots, T_x, \dots, T_N)$. Each of the components T_x is not a moment, while representing the start time T_{sx} and the end time T_{ex} of the resource allocation. T_{sx} represents the time at which the first service or application begins to be allocated during the resource allocation process. Similarly, T_{ex} represents the final end time of the resource allocation for the service.

Here, the corresponding time changes in the resource allocation process are also been considered. The process of allocating resources to each service will have a fixed processing time and will be described as $TI = (TI_1, TI_2, \dots, TI_x, \dots, TI_N)$. Each component TI_x represents the initial time TI_{sx} at which the resource allocation is just started and the end time TI_{ex} after the allocation is completed. TM_x is equal to the center position of two time points, where TI_x can be determined by the center time TM_x , denoted as $TI(TM_x) = TI_x$. Assuming that the optimal allocation time of each service or application in the resource configuration process is $t = (t_1, t_2, \dots, t_x, \dots, t_N)$, the pseudo code shown in Table 1 can be used to obtain the solution of optimal configuration time window:

Table 1 The algorithm of optimizing configuration windows**Algorithm** Optimizing configuration windows

Input: the priority sequence of applications, resource allocation scheme, the time for each service to allocate resources;

Output: the optimal allocation time $t = (t_1, t_2, \dots, t_x, \dots, t_N)$;

```

1: for (each  $S_x$  in  $SE$ )
2:   find F;
3:   for (each  $F_x$  in F)
4:     find  $T_x, TI_x$ ;
5:     get t make  $p_{RS}(t)$  is maximal when  $t \in T_x$ ;
6:     if ( $TI(t) \subseteq T_x$ )
7:        $t_x = t$ ;
8:     else
9:       set  $TI_{sx} = T_{sx}, tm = TM_x, tem = p_{RS}(tm)$ ;
10:      get tm, tem;
11:      set  $TI_{ex} = T_{ex}, tn = TM_x, ten = p_{RS}(tn)$ ;
12:      get tn, ten;
13:      if ( $tem \geq ten$ )
14:         $t_x = tm$ ;
15:      else
16:         $t_x = tn$ ;
17:      end if
18:    end if
19:  end for
20: end for

```

4 Simulation Experiments and Analysis

So far, the existing methods for evaluating survivability of network systems can be roughly divided into two categories: qualitative assessment and quantitative assessment of survivability. For qualitative assessment, the most classic is the SNA proposed by Knight etc. [16], which can also be used to design and guide survivable network systems. The SNA method proposes to evaluate the survivability of network systems from the three perspectives of intrusion prevention, identification attack and recovery service. Later, the quantitative assessment of survivability has gradually attracted the attention of researchers, and many research results have been achieved. In the experiments in this section, the simulation experiments and results of the proposed algorithm are carried out by using the index based on semi-Markov process proposed by Bai [17] and the gray correlation analysis index based on information entropy difference proposed by Wang [18].

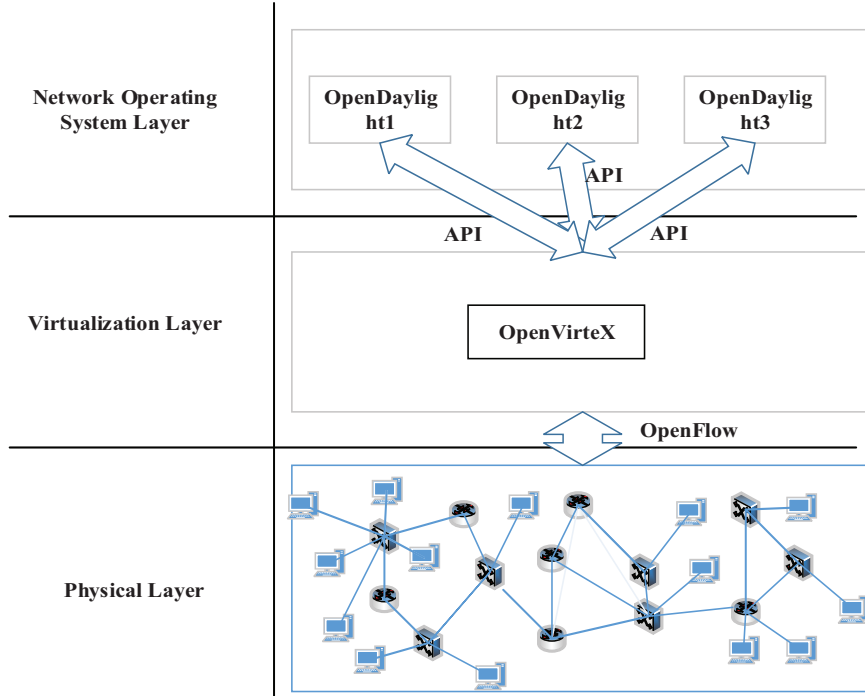


Figure 1 Experimental verification platform architecture under SDN network.

The experiments are conducted in a SDN network environment using a three-layer verification platform for simulation experiments, as shown in Figure 1. The physical layer uses Mininet to define a medium-sized SDN; the virtualization layer uses OpenVirteX software to create and manage multiple virtual networks; the network operating system layer controls the entire network by using the data flow table sent by the OpenDaylight controller. The underlying physical network is designed to have a topology of 50 nodes and approximately 25 links. The probability of connection between each pair of physical nodes is set to 0.5, and the forwarding resources and control resources of each node and the bandwidth resources of the physical link are subject to a uniform distribution of [50, 100] respectively. The time window is used to represent the arrival and departure of the service. Each time window is 100 time units. Within a time window, set the number of service arrived to obey a Poisson distribution with the mean of 4, and the survival time of the service is set as the exponential distribution with the mean of 10 time windows.

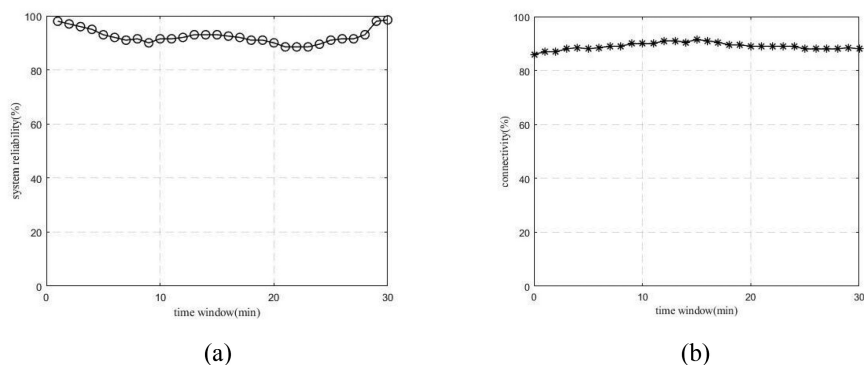


Figure 2 Trends of the system survivability index with time.

Firstly, the various services and application requirements requested by the user are sorted according to the utility function. It is the order in which the network resources are configured. Then the resource allocation problem is optimized according to the constraints, and the configuration time window is also optimized. The simulation experiment results are shown in Figure 2(a). The system reliability index analyzes and judges the survivability of network system from the perspective of the quality attribute of the system. As can be seen from Figure 2(a), using the algorithm proposed in this paper to configure the resources, the reliability of the system can be maintained at around 90% even under attacks. The system is still able to complete its critical services, which guarantees high reliability. The connectivity indicator emphasizes the interconnectivity of the physical and logical structures of software systems. According to this indicator, the system can tolerate faults in the event of human damage, failure, or major accidents. As shown in Figure 2(b), the system connectivity obtained by the simulation experiment can well guarantee the quality of user service under unexpected conditions.

The three indicators of bandwidth utilization, average response time, and transmission rate are important attributes for evaluating system performance. By collecting data, judging descriptions, and then normalizing the above indicators, the survivability of the final system can be reasonably evaluated. According to the simulation experiment results, it can be found that the algorithm proposed in this paper is assessed based on the evaluation method. As shown in Figure 3, the results show that the survivability of the system does not change much over time, and remains at a high level. In other words, this means that the algorithm studied in this paper can improve the survivability of the system.

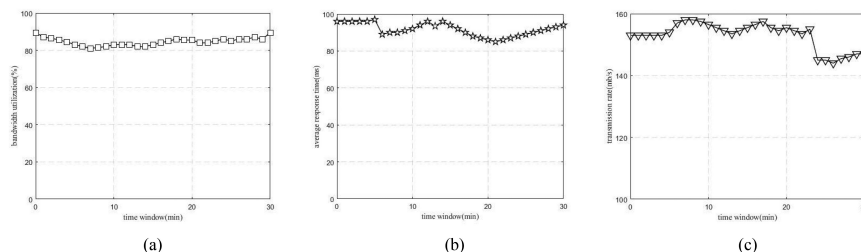


Figure 3 Trends of the other system index with time.

5 Conclusions

Aiming at maximizing the satisfaction of user's QoS, the automatic configuration method for system resources is studied for the user service that the system can provide without manual intervention under normal circumstances. The method makes use of the perceived internal and external environment of the system and its own state, and learns from the self-configuration idea in the cognitive network. By introducing the utility function in the consumption domain as an indicator, a self-configuration method for cognitive network is constructed to solve the resource allocation, and the resources are dynamically configured. Through simulation experiments, it is verified that the algorithm has greatly improved the system reliability, connectivity, bandwidth utilization, average response time, and transmission rate for the traditional resource allocation scheme. Thus, the ability that the system intelligently adapting to the environment is improved, meeting the needs of survivability growth. The optimal self-configuration algorithm for cognitive networks studied in this paper is aimed at the network topology with small scale. If the network topology is too complicated, the complexity of the problem will be high, which makes the algorithm inefficient. In the future work, the topology complexity of the network will be taken into account to improve the efficiency of the algorithm.

Acknowledgements

This present research work was supported by the National Natural Science Foundation of China (61403109, 61202458), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20112303120007), the Heilongjiang Natural Science Foundation (F2017021), the Scientific Research Fund of Heilongjiang Provincial Education Department (12541169)

and the Specialized Research Fund for Scientific and Technological Innovation Talents of Harbin (2016RAQXJ036).

References

- [1] Westmark VR. A definition for information system survivability. Proceedings of the 37th Hawaii Internal Conference on System Sciences, 2004: 2086–2096.
- [2] Loredana A, Daniele T. Stochastic optimization of cognitive networks. IEEE Transactions on Green Communications and Networking, 2017, 1(1): 40–58.
- [3] Francesco C, Romano F, Andrea T. Performance evaluation of an adaptive channel allocation technique for cognitive wireless sensor networks. IEEE Transactions on Vehicular Technology, 2017, 66(6): 5351–5363.
- [4] Mustafa HY, Hüseyin A. Resource allocation with partially overlapping filtered multitone in cognitive heterogeneous networks. IEEE Communications Letters, 2016, 20(5): 962–965.
- [5] Alexander F, Uwe M, Volker L. A concept for self-configuration of adaptive sensor and information fusion systems. Proceedings of the 21st International Conference on Emerging Technologies and Factory Automation, 2016: 1–4.
- [6] Kaindl H, Vallee M, Arnautovic E. Self-representation for self-configuration and monitoring in agent-based flexible automation systems. IEEE Transactions on Systems Man & Cybernetics Systems, 2013, 43(1): 164–175.
- [7] Feng G, Wang H, Li B, et al. Dynamic self-configuration of user QoS for next generation network. Proceedings of the 6th IFIP International Conference on Network and Parallel Computing, 2009: 80–85.
- [8] Anthony R, Pelc M, Ward P, et al. A run-time configurable software architecture for self-managing systems. Proceedings of the 2008 International Conference on Autonomic Computing, 2008: 207–208.
- [9] Msadek N, Kiefhaber R, Ungerer T. A trustworthy fault-tolerant and scalable self-configuration algorithm for organic computing systems. Journal of Systems Architecture, 2015, 61: 511–519.
- [10] Yaqub MF, Gondal I, Kamruzzaman J. An adaptive self-Configuration scheme for severity invariant machine fault diagnosis. IEEE Transactions on Reliability, 2013, 62(1): 160–170.

- [11] Peng M, Liang D, Wei Y, et al. Self-configuration and self-optimization in LTE-advanced heterogeneous networks. *IEEE Communications Magazine*, 2013, 51(5): 36–45.
- [12] Mohamed L, Michael B, Anne F. Self-configuration mechanisms for SDN deployment in wireless mesh networks. *Proceedings of the IEEE 18th International Symposium on World of Wireless, Mobile and Multimedia Networks*, 2017: 1–4.
- [13] Hu JJ, Guan HN, Wei J, etc. A performance model-based self-configuration framework. *Journal of Software*, 2007, 18(9): 2117–2129.
- [14] Lin YM, Cai GY, Li Y. Formal modeling and analysis of policy-based self-configuration system. *Computers and Modernization*, 2008, (2): 63–66.
- [15] Jia Y, Zhang XZ. Research on adaptive configuration framework for LQNM-based database in cloud environment, *Computer Applications and Software*, 2015, 32(9): 49–53.
- [16] Knight JC, Strunk EA, Sullivan KJ. Towards a rigorous definition of information system survivability. *DARPA Information Survivability Conference and Exposition*, Washington, USA, 2003.
- [17] Bai B. *Survivable network system framework and quantitative evaluation of network survivability*. ChengDu: University of Electronic Science and Technology of China, 2010: 45–68.
- [18] Wang YL, Zhang Y, Yu ZW, etc. Research on quantitative of network storage system survivability. *Computer Engineering*, 2010, 36(4): 33–37.

Biographies



Jian Wang received the PhD degree in computer applications in 2009 from Harbin Engineering University, China. She is currently a professor and doctoral supervisor in school of computer science and technology, Harbin

University of Science and Technology. Her research interests include cognitive network, survivability, trusted computing and crowd sensing.



Guosheng Zhao received the PhD degree in computer applications in 2009 from Harbin Engineering University, China. He is currently a professor and master supervisor in school of computer science and information engineering, Harbin Normal University. His research interests include cognitive network, trusted computing and IoT security.



Zhongnan Zhao received the PhD degree in computer applications in 2017 from Harbin University of Science and Technology, China. He is currently an associate professor in school of computer science and technology, Harbin University of Science and Technology. His research interests include security situation awareness, intelligent computing and fault-tolerance.

Zhixin Li is currently a postgraduate in school of computer science and technology, Harbin University of Science and Technology. Her main research interests include survivability, cognitive computing and SDN.