SDN Controller Scheduling Decision Mechanism Based on Mimic Defense Affects Safety

Desheng Zhang and Lei Chen(✉)

Jiangsu Province Key Laboratory of Intelligent Industry Control Technology, Xuzhou University of Technology, Xuzhou 221018, China
chenlei@xzit.edu.cn

Abstract. Nowadays, with complex network structures, diversified network equipment, and diversified network applications, more and more complex functions and requirements have led to traditional network management methods that are no longer competent for current networks. The emergence of SDN relies on the characteristics of separation of data forwarding and control, and has a huge advantage in network management in the new era, but SDN also faces security threats that have not been experienced by traditional networks. Since the control of the SDN network is centralized, the security of the SDN controller is particularly important. This article uses mimic defense to ensure the security of the controller, focusing on the security impact of different decisions of the decision maker in the mimic defense system, and optimizes the decision-making method, Found the decision-making method that can maintain the safe operation of the system for the longest time, and verified it through simulation experiments.

Keywords: Mimic defense · Software defined network · Controller

1 First Section

With the rapid development of computer technology, many emerging technologies have emerged, such as some big data [1, 2], Internet of Things [3, 4], industrial Internet [5, 6], cloud computing [7, 8] and other excellent Technology, these technologies have even changed our cities [9, 10], they also put forward higher requirements for the basic network system of the information highway, and through the development of mobile Internet [11, 12], network With the continuous expansion of scale, the variety of network equipment, and the application of new network technologies to more environments [13, 14], it is only a method of manual configuration by network practitioners to achieve each end-to-end stable link [15, 16] It is difficult to guarantee. Faced with these challenges, software-defined networking SDN [17, 18] ushered in rapid development.

SDN defines and controls the network in the form of software programming, and realizes the separation of its control plane and forwarding plane. SDN has mature applications in different places [19, 20]. Due to the characteristics of SDN, SDN can be combined with many technologies. Similar artificial intelligence technologies [21, 22], the

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 370, pp. 278–288, 2021.
https://doi.org/10.1007/978-3-030-72795-6_22

key to achieving many functions lies in the control of traffic [23, 24] With traffic status, routing can be better processed Forward [25, 26] to achieve traffic prediction [27, 28]. SDN is the trend of future network development and will bring qualitative changes to the entire Internet.

Mimic defense technology [29] is a dynamic discrete superposition [30] structure based on the endogenous security mechanism of cyberspace. The mimic defense technology integrates multiple active defense [31, 32] elements as a whole. The realization of mimic defense needs to be established the dynamic redundancy architecture realizes the dynamics, randomness, and diversity of the system while ensuring the replacement of system functions. Mimic technology can promote the improvement of the security of SDN controllers [33, 34]. The actual operation can be through multiple alternative controllers, while receiving the entire winding state and updating it in real time, but when there is a routing request that needs to be issued flow rules, It is necessary to go through the decider first, and according to a certain decision rule, send the theoretically safest flow rule to the network device. One more single controller is just the basis for mimic defense to improve the system's defense capabilities. The core of the real feasible and reasonable play of multiple alternate controllers lies in whether the strategy of the decider is reasonable.

2 Problem Introduction

The arbiter used in mimic defense has a variety of decision-making strategies, such as majority selection, rotation selection, random selection and so on. No matter what the strategy is adopted, to ensure the normal function of the premise, as far as possible to enhance the whole system security body, will be the same pursuit. We build a basic mimic defense structure at the SDN control layer. There are n heterogeneous controllers $\{C_1, C_1, C_2, C_3 \cdots C_n\}(n \geq 2)$, Run on k different operating systems $\{OS_1, OS_2, OS_3 \cdots OS_n\}$, all the controllers are connected to a same arbitrator A of 0. The arbiter N0 through Openflow [35] controls complex network $[\![\text{Net}]\!]\_0$ protocols below, There are m switches in the network N0 $\{S_1, S_2, S_3 \cdots S_n\}(m \geq 1)$, shown in Fig. 1.

In this case For Net_0 each switch S_m, it is logically linked with a single controller, meridians Openflow protocol controller normal exchange of link information, wherein mimicry defense portion of the link switch, either by The dedicated link can link the entire system to the Net_0 inside of the network. All n controllers and operating systems running on the n controllers may not necessarily be all different, but it is necessary to ensure that the combinations of controllers and operating systems are not the same. For example: the same controller C_a can be placed in the operating system, respectively $\{0S_a, 0S_b\}$, form two combinations $\{(0S_a, C_a), (0S_b, C_a)\}$, with corresponding operating systems can run different controller to form a new combination.

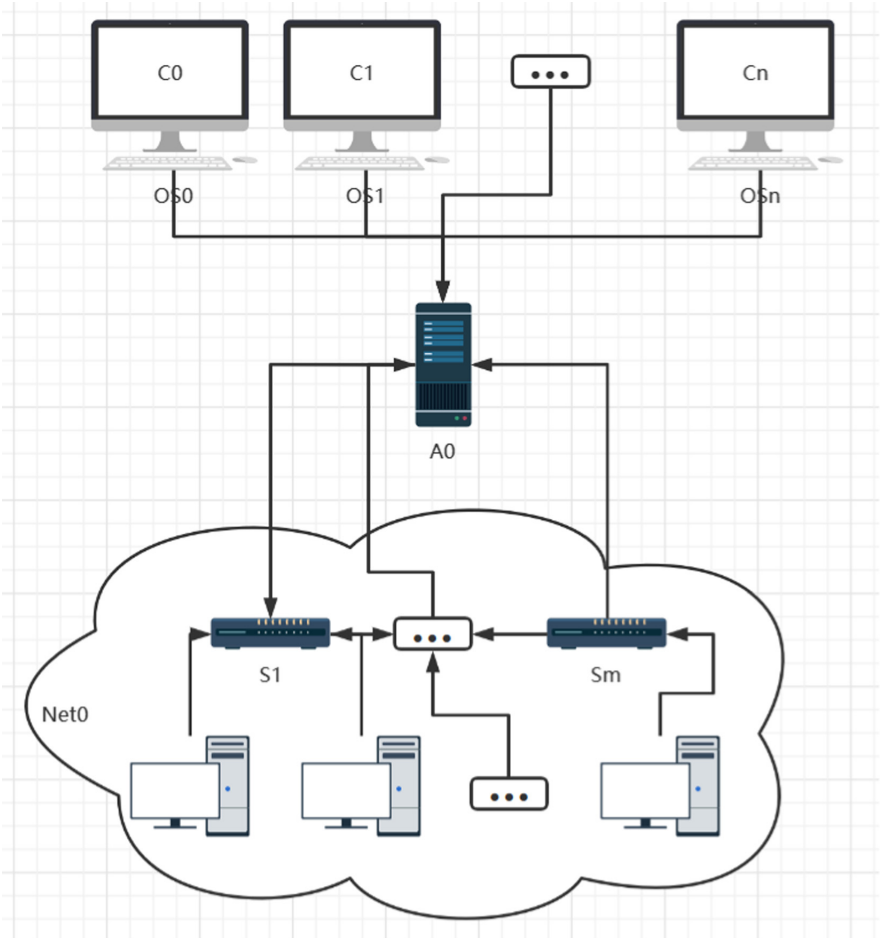

Fig. 1. Based defense mimic the structure.

2.1 Safety of a Single Controller

First of all, we must understand that the general security of the controller can be ensured by reasonable security configuration and timely security patches. However, when faced with advanced sustainable threats, the security of a single controller is difficult to ensure. During the attack, the probability P of a single controller being compromised will increase with the passage of time, and will eventually approach 1 infinitely. The probability of the controller being broken can be approximated to an exponential distribution, and the distribution function of the probability P of being broken over time T is as follows:

$$P = F(t) = \begin{cases} 1 - e^{-\lambda t}, t \geq 0 \\ 0, t < 0 \end{cases} \tag{1}$$

Among them, λ is affected by the product quality q and the attacker's ability x:

$$\lambda = K\frac{x}{q} \tag{2}$$

2.2 Security Policy Decision Maker

There are many selection methods that the selector can use, such as majority selection, alternate selection, random selection, and so on. Under the premise that a single controller must be breached within a certain period of time, no matter whether you choose to execute in turn or execute randomly, there must be a time set $T = \{T_1, T_2, T_3 \cdots\}$, and the same controller is running. Within these time periods, the entire system A pure SDN network is no different, and an attacker can continue to attack a single controller after it is taken. The entire system will be controlled by the attacker within time T. In this case, the attacker can use or destroy the entire link system through the compromised controller even in a short period of time. Single controller design break time T_{single}, the entire system is compromised time T_{system} is:

$$T_{system} = T_n(n = \min(\{n| \sum_{i=1}^{i=n} len(T_i \geq T_{single}\})) \tag{3}$$

And a single controller is compromised as a probability P_{single}, the probability of the entire system is compromised P_{system} is:

$$P_{system} = P_{single} \tag{4}$$

We can be seen that this is not a particularly reliable choice.

At present, the most mainstream choice of mimic defense is the majority option, that is, each time the flow table is issued, the arbiter compares all the results, and finally selects the result of the majority controller. This allows a single controller to be compromised and does not affect the overall operation of the system. Even if an attacker takes down a single controller, it does not affect the overall operation of the system. If the attacker wants to control the entire system, in a total of n controllers under the premise, the attacker would need to take $k = \lceil \frac{n}{2} \rceil$ controllers. The time when the entire system was breached T_{system} is:

$$T_{system} = k \times T_{single} = \lceil \frac{n}{2} \rceil \times T_{single} \tag{5}$$

In most cases, the probability of the system being compromised P_{system} is:

$$P_{system} = \sum_{k=\lceil \frac{n}{2} \rceil}^{n} C_n^k {P_{single}}^k (1 - P_{single})^{n-k} \tag{6}$$

Can be seen from the equation that, when the safety factor is high enough based controller, with the safety factor of the overall system increases the number of controllers will rise quickly when the controller itself is excellent enough mass, to be a plurality of states defense Redundancy will have a very limited improvement in the entire system. When the probability of the controller being compromised is higher than 0.5, the entire multiple redundant components will have a reverse effect and reduce the safety factor of the entire system.

2.3 Handling of Abnormal Controller

For most selected cases, when a single controller sends out wrong results multiple times, we can judge that this controller is abnormal and may have been controlled by an attacker. You can choose to save the scene for digital forensics for the controller first, and then restore it by mirroring. Time attacks and handling exceptions at different times will bring different security effects. Because some controllers will be shut down during operation, the safety factor of the entire system changes with time. In order to improve the overall safety of the system as much as possible, the abnormal controller is shut down as late as possible. The safety factor changes with time as shown in the figure below. The time to break a single controller is, the time to T_{single}, the time to close the i-th controller should not be less than $\lfloor \frac{n}{2} t \rfloor$. This can significantly increase the time the system lasts to defend against attacks. However, if it is just a simple mirror restoration, the time cost for an attacker to use the original vulnerability to compromise the controller again will be greatly reduced. The time cost of the first attack is t_0, the time cost of the second attack is t_1, and the time cost of the nth attack is t_n. The relationship between them should be:

$$t_n = \sqrt[m]{t_{n-1}} \tag{7}$$

Among them, m is a constant, which means that the cost of reattacking the controller after being attacked by a vulnerability will quickly drop to a constant level. Therefore, the number of restarts should be limited and at an appropriate time.

The cost of closing the controller directly or saving the image will be high. There is also a more cost-effective control method, that is, the selector is not a simple choice for the majority, and the controller can be weighted. Depending on the controller settings for different controller associated weights W, issued in all controllers m result set {R1, R2, R3···}, wherein the controller {nx, ny, nz···} The issued result is R1, and its corresponding weight is {wx, wy, wz···}. The weight of a single result, that is, W_i is:

$$W_i = \sum w_k, \; w_k \text{ is the weight of the controller } n_k \in \{\text{Output is } R_i\} \tag{8}$$

The final result R selected by the selector is:

$$R = R_{\max(w1,w2,w3\cdots wm)} \tag{9}$$

As a result, the controller updates the weight of the controller according to the result after each command:

$$\begin{cases} wi = wi + 1, n_i \; result = finall \; result \\ wi = \sqrt{wi}, n_i result \; \neq \; finall \; result \end{cases} \tag{10}$$

It is also possible to achieve results equivalent to shutdown.

3 Simulation Results and Points Analysis

3.1 Experimental Environment

In order to simply test the previous theories and solutions, we build a simple SDN test structure, try VMware to install multiple operating systems, such as Windows server,

Ubuntu, Debian, etc. Is not mounted in different virtual machines to make the same control device, such as RYU, Opendaylight, the Floodlight etc., using Mininet [36] analog SDN switches and Python write arbiter and process the data. The SDN network has no special requirements, as long as it is not too simple, the process of the controller being compromised can indicate the abnormality of the controller by actively asking the controller to send an error flow table.

3.2 Experimental Evaluation

Among the most choices, the probability of each controller being compromised by the attacker is p. The influence of probability p and the number of controllers on the entire system is shown in Fig. 2:

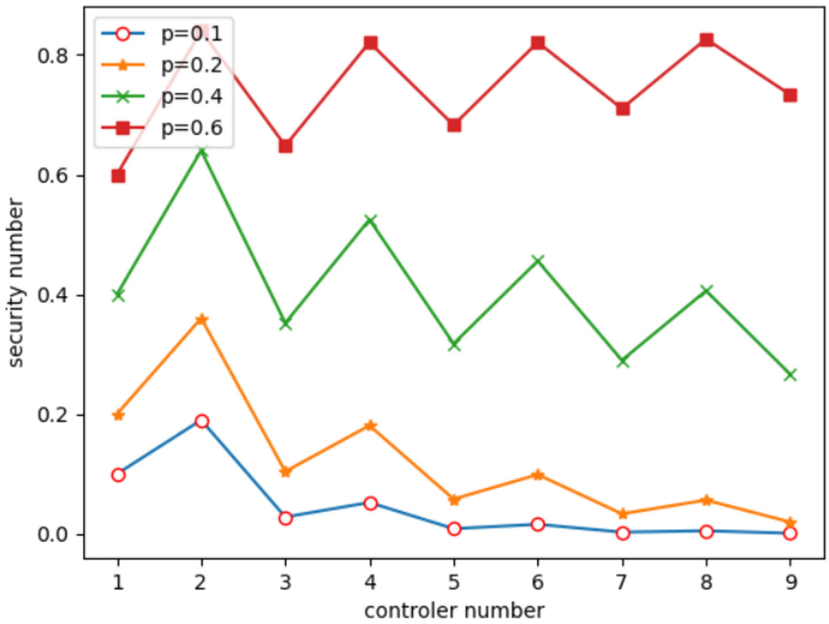

Fig. 2. The relationship between controller and system security under different p conditions.

It can be seen that when the basic safety factor of the controller is high enough, the safety factor of the entire system will increase rapidly as the number of controllers increases, and when the quality of the controller itself is not good enough, multiple redundancy of mimic defense The improvement of the entire system is very limited. When the probability of the controller being compromised is higher than 0.5, the entire multiple redundant components will have a reverse effect, reducing the safety factor of the entire system.

The average time it takes for the attacker to obtain the control of a single controller is T, and the time spent to control the entire system with the authority is shown in Fig. 3, which shows the non-processing of exceptions and the previous strategy. Shut down

multiple controllers. At the same time, Fig. 4 shows the trend of the controller safety factor over time.

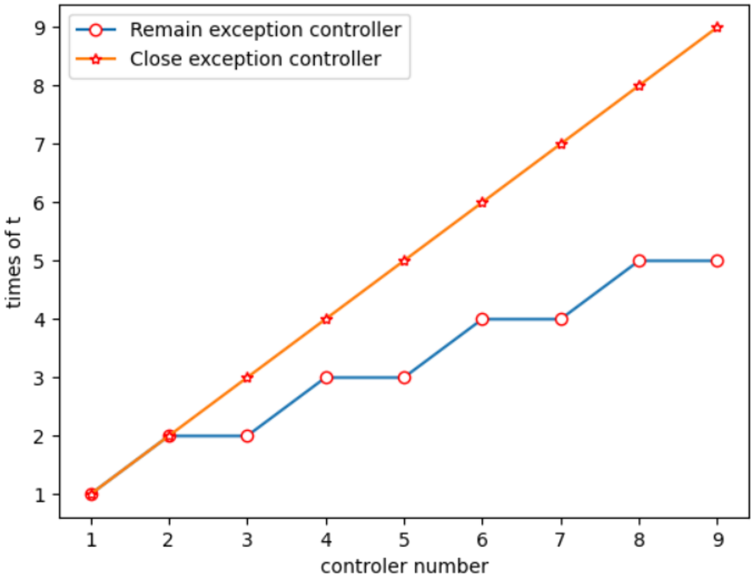

Fig. 3. The time spent to control the entire system.

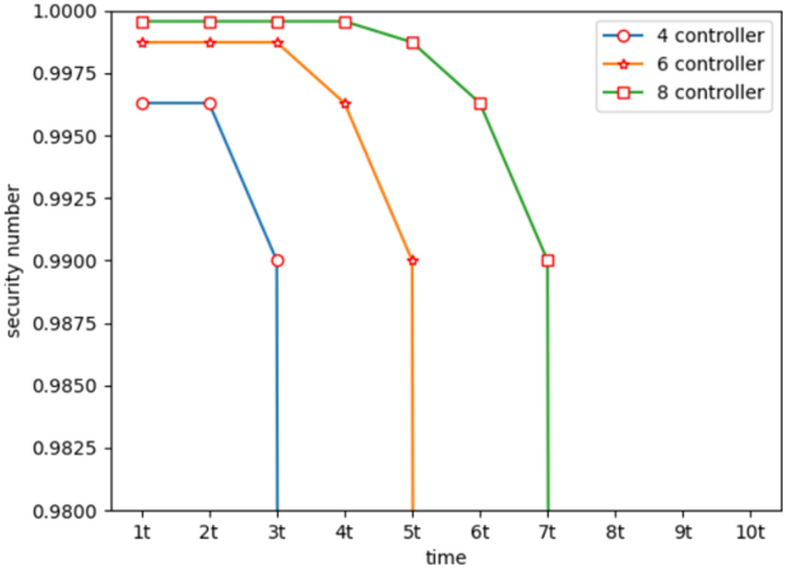

Fig. 4. The trend of the controller safety factor.

It can be seen that the method of gradually turning off the controller can not only effectively extend the time for the system to resist attacks, but also because it is turned off accordingly, it can also maintain the overall safety factor of the system as much as possible.

Regarding the weight processing ruling method, we send erroneous messages at regular intervals to indicate that the system is abnormal. At the initial weight of 5000, there are four controllers for mimic defense. After each abnormal message is sent three times, we can see the result as follows Fig. 5 weight changes. And the more intuitive weight ratio is shown in Fig. 6.

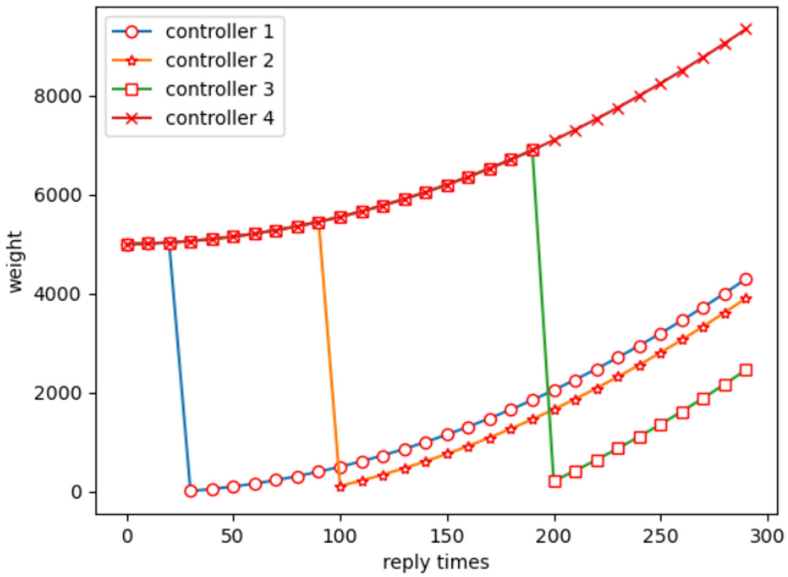

Fig. 5. The changing trend of the controller.

We can be seen that only after 30 times, 100 times and 200 times, the controller only performs three abnormal operations and its weight drops rapidly, which is equivalent to "temporarily shutting down" the controller.

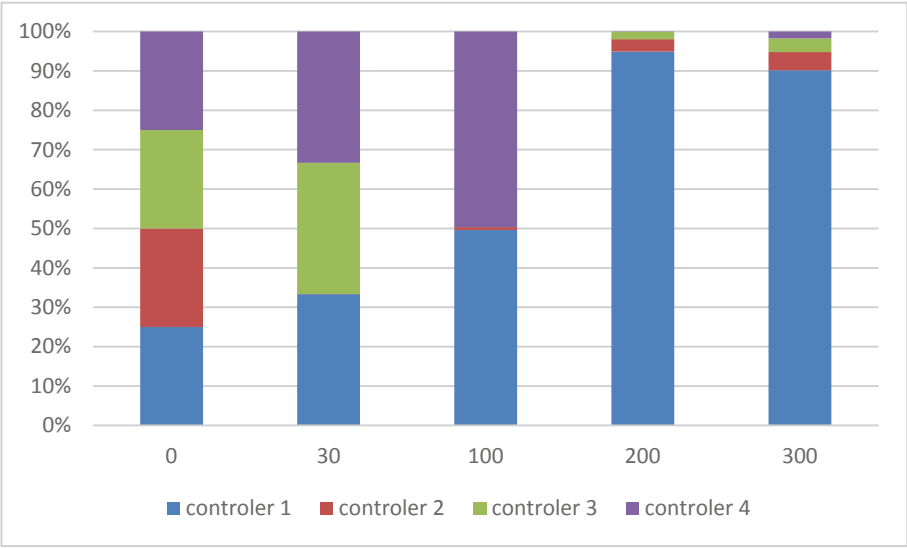

Fig. 6. Trend of weight change.

4 Conclusion

Mimic defense can well defend against known and unknown attacks. We have systematically analyzed the problems faced by the controller's mimic defense technology, and quantified the relevant conditions to find the relevant optimal strategy, so that we are all Enough to give full play to the defensive effect of the mimic defense system, improve the overall security of the system, and finally verify our relevant conclusions through simulation experiments, and also give a more intuitive demonstration of the effect of the mimic defense system.

References

1. Malhotra, S., Doja, M.N., Alam, B., Alam, M.: Bigdata analysis and comparison of bigdata analytic approches. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, pp. 309–314 (2017). https://doi.org/10.1109/CCAA.2017.8229821
2. Jiang, D., Wang, Y., Lv, Z., Qi, S., Singh, S.: Big data analysis based network behavior insight of cellular networks for industry 4.0 applications. IEEE Trans. Ind. Inf. **16**(2), 1310–1320 (2020)
3. Gupta, A.K., Johari, R.: IOT based electrical device surveillance and control system. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, pp. 1–5 (2019). https://doi.org/10.1109/IoT-SIU.2019.8777342
4. Park, D., Bang, H., Pyo, C.S., Kang, S.: Semantic open IoT service platform technology. In: 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, pp. 85–88 (2014). https://doi.org/10.1109/WF-IoT.2014.6803125

5. Nagpal, C., Upadhyay, P.K., Shahzeb Hussain, S., Bimal, A.C., Jain, S.: IIoT based smart factory 4.0 over the cloud. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, pp. 668–673 (2019). https://doi.org/10.1109/ICCIKE47802.2019.9004413
6. Jiang, D., Wang, Y., Lv, Z., Wang, W., Wang, H.: An energy-efficient networking approach in cloud services for IIoT networks. IEEE J. Sel. Areas Commun. **38**(5), 928–941 (2020)
7. Bahrami, M.: Cloud computing for emerging mobile cloud apps. In: 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, San Francisco, CA, pp. 4–5 (2015). https://doi.org/10.1109/MobileCloud.2015.40
8. Linthicum, D.S.: Connecting fog and cloud computing. IEEE Cloud Comput. **4**(2), 18–20 (2017). https://doi.org/10.1109/MCC.2017.37
9. Abu-Matar, M., Mizouni, R.: Variability modeling for smart city reference architectures. In: 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, pp. 1–-8 (2018). https://doi.org/10.1109/ISC2.2018.8656967
10. Jiang, D., Zhang, P., Lv, Z., et al.: Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet Things J. **3**(6), 1437–1447 (2016)
11. Bai, B., Guo, Z.: Dynamic complexity of mobile internet business ecosystem. In: 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, pp. 502–506 (2017). https://doi.org/10.1109/ICSAI.2017.8248344
12. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **7**(1), 80–90 (2020)
13. Jiang, D., Huo, L., Lv, Z., Song, H., Qin, W.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
14. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **220**, 160–169 (2017)
15. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. **7**(1), 507–519 (2020)
16. Qi, S., Jiang, D., Huo, L.: A prediction approach to end-to-end traffic in space information networks. Mobile Netw. Appl. (2019)
17. Prajapati, A., Sakadasariya, A., Patel, J.: Software defined network: future of networking. In: 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, pp. 1351–1354 (2018). https://doi.org/10.1109/ICISC.2018.8399028
18. Raychev, J., Hristov, G., Kinaneva, D., Zahariev, P.: Modelling and evaluation of software defined network architecture based on queueing theory. In:, 2018 28th EAEEIE Annual Conference (EAEEIE), Hafnarfjordur, pp. 1–5https://doi.org/10.1109/EAEEIE.2018.8534289
19. Theodorou, T., Mamatas, L.: CORAL-SDN: a software-defined networking solution for the internet of things. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, pp. 1–2 (2017). https://doi.org/10.1109/NFV-SDN.2017.8169870
20. Tantayakul, K., Dhaou, R., Paillassa, B.: Mobility management with caching policy over SDN architecture. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, pp. 1–7 (2017). https://doi.org/10.1109/NFV-SDN.2017.8169830
21. Huo, L., Jiang, D., Qi, S., et al.: An AI-based adaptive cognitive modeling and measurement method of network traffic for EIS. Mobile Netw. Appl. (2019)
22. An AI based Approach to Secure SDN Enabled Future Avionics Communications Network Against DDoS Attacks
23. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)

24. Huo, L., Jiang, D., Lv, Z., et al.: An intelligent optimization-based traffic information acquirement approach to software-defined networking. Comput. Intell. 1–21 (2019)
25. Wang, F., Jiang, D., Qi, S.: An adaptive routing algorithm for integrated information networks. China Commun. **7**(1), 196–207 (2019)
26. Raza, S.M., Thorat, P., Challa, R., Choo, H., Kim, D.S.: SDN based inter-domain mobility for PMIPv6 with route optimization. In: 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, pp. 24–27 (2016). https://doi.org/10.1109/NETSOFT.2016.7502436
27. Abadi, A., Rajabioun, T., Ioannou, P.A.: Traffic flow prediction for road transportation networks with limited traffic data. IEEE Trans. Intell. Transp. Syst. **16**(2), 653–662 (2015). https://doi.org/10.1109/TITS.2014.2337238
28. Wang, Y., Jiang, D., Huo, L., Zhao, Y.: A new traffic prediction algorithm to software defined networking. Mobile Netw. Appl. (2019)
29. Ma, B., Zhang, Z.: Security research of redundancy in mimic defense system. In: 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, pp. 2910–2914 (2017). https://doi.org/10.1109/CompComm.2017.8323064
30. Yoon, G., Lee, S., Kwon, D., Kwon, S., Park, Y.: RAPIEnet based redundancy control system. In: 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, pp. 140–145 (2011)
31. Research on the active defense security system based on cloud computing of wisdom campus network
32. Xia, J., Cai, Z., Hu, G., Xu, M.: An active defense solution for ARP spoofing in OpenFlow network. Chinese J. Electron. **28**(1), 172–178 (2019). https://doi.org/10.1049/cje.2017.12.002
33. Tatang, D., Quinkert, F., Frank, J., Röpke, C., Holz, T.: SDN-guard: protecting SDN controllers against SDN rootkits. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, pp. 297–302 (2017). https://doi.org/10.1109/NFV-SDN.2017.8169856
34. Boukria, S., Guerroumi, M., Romdhani, I.: BCFR: blockchain-based controller against false flow rule injection in SDN. In: 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, pp. 1034–1039 (2019). https://doi.org/10.1109/ISCC47284.2019.8969780
35. Qureshi, S., Braun, R.: Mininet topology: mirror of the optical switch fabric. In: 2019 29th International Telecommunication Networks and Applications Conference (ITNAC), Auckland, New Zealand, pp. 1–6https://doi.org/10.1109/ITNAC46935.2019.9078014
36. Pereíni, P., Kuzniar, M., Kostic, D.: OpenFlow needs you! A call for a discussion about a cleaner OpenFlow API. In: 2013 Second European Workshop on Software Defined Networks, Berlin, pp. 44–49 (2013).https://doi.org/10.1109/EWSDN.2013.14