Research on QoS routing method based on NSGAII in SDN

Daoquan Li¹, Xue Wang¹, Yingnan Jin¹ * and Haoxin Liu¹
¹ Department of Computer Engineering, Qingdao University of Technology, Qingdao, Shandong, 266520, China
*Corresponding author’s e-mail: jdh@qtech.edu.cn

Abstract. Software Defined Network (SDN) makes routing decisions according to the shortest path. This routing scheme does not consider link information such as link residual bandwidth, forwarding delay, packet loss rate and so on. In the case of large data flow, it is easy to cause network congestion, thus reducing the network quality of service assurance. For this reason, using the ability of SDN network controller to monitor the underlying network information and control the route forwarding, we try to set multiple objective functions according to different QoS parameters, and use the non-dominated sorting multi-objective optimization genetic algorithm (NSGA II) based on the elite strategy to make routing decisions. Compared with other routing algorithms, the proposed multi-objective optimization genetic algorithm reduces forwarding delay and packet loss rate, so can effectively prevent network congestion.

1. Introduction
Software Defined Network (SDN) [1] is a new Network mode, with a centralized and programmable control plane, which can flexibly control the route and timely meet the fine-grained traffic needs. The deployment of SDN enables network operators to carry out efficient online routing optimization according to the current situation of the network [2]. The experimental results in the data center network supported by Microsoft [3] and Google [4] verify that the SDN network can achieve nearly optimal performance in throughput and link utilization through the implementation of routing optimization.

The Quality of Service (QoS) should be fully considered when researching the routing strategy of SDN. There are two types of QoS, one is parameterized QoS and the other is priority QoS. Parameterized QoS is a kind of strict QoS, that is the SDN controller makes routing decisions based on the information such as available bandwidth and link delay in the underlying network. However, Priority QoS is a weak priority, which only requires different routing priorities between different traffic categories.

2. Related works
Literature [5] proposed a routing strategy based on genetic algorithm, which can guarantee the quality of video traffic transmission. The core idea of this strategy is to identify video packets through specific ports and special fields in IP data headers and find the optimal path for them globally by using genetic algorithm. However, it does not consider the impact of bandwidth, delay and other parameters on the path selection in the network. Literature [6] used the queue control resource allocation scheme on the OpenFlow protocol V1.3 [7], which can simultaneously solve the bandwidth demand of terminal hosts on multiple paths. In the aspect of multi-path routing, this method only meets the bandwidth demand of customers and guarantees the economical utilization of switch resources, but does not consider
other QoS parameters in the network. When the network data flow is large, this routing method is easy to cause the link congestion, which leads to the increase of forwarding delay and other network problems.

Literature [8] proposed a routing mechanism based on business division in SDN network. This mechanism distinguishes the business types of data flow in the network according to the ToS field, and selects different paths for different business flows among multiple alternative paths to ensure the requirements of different businesses, which can effectively avoid the problem of limited resources in the network of blind competition between different traffic flows. However, the method of choosing the path is ambiguous to some extent because the weight is calculated by setting the parameter artificially. Literature [9] restricted the crossover and mutation operations of genetic algorithm to reduce the computational complexity of the algorithm so that it can choose the best forwarding path adaptively according to the dynamic changes of network resources. This method is experimentally tested from two aspects: elephant flow and mouse flow [10],[11]. The results show that the proposed routing algorithm can effectively reduce delay and packet loss rate and improve network performance. However, the fitness function formula designed in this literature also needs to set the weight artificially, which transforms the multi-objective optimization problem into the single-objective optimization genetic algorithm.

Based on the above literature research, in this paper, network parameters such as link residual bandwidth, transmission delay and packet loss rate are comprehensively considered, and a kind of non-dominated sorting based on the strategy of elite QoS routing method of multi-objective optimization genetic algorithm (NSGAII_QoS) in SDN network is proposed, which can set different target functions for multiple QoS parameters in the network, and automatically find the optimal path of multiple targets without setting parameters artificially.

3. Model of NSGAII_QoS

The concentricity and programmability of SDN controller provide the foundation for QoS routing. This article add a multi-objective optimization genetic algorithm based on the elitist strategy of routing module (NSGAII_QoS) to the SDN controller, which contains three modules: network monitoring module, network awareness module, and network reconfiguration module.

3.1. System structure

In the design of routing strategy, the controller needs to acquire the information of network resources in real-time, including the number of switch nodes, network topology information, link resource information, etc. Therefore, this paper adds corresponding modules in the RYU controller to complete their respective functions, and the specific system structure is shown in Figure 1.

First, the Hello module sends a Hello message to the switch to establish a secure channel. Then the Features module issues a Features request for each switch to obtain the maximum link bandwidth associated with each port on the switch. When the packet of a new stream arrives in the SDN network, the access switch sends the packet_in message containing the request of the new packet forwarding rule to the controller. After that, the controller issues the forwarding rule to the access switch via the packet_out message. The Flow mod module can set the routing rules of the switch for the shortest transport path of traffic. In terms of information collection, the controller sends a Flow Stats request to the access switch to obtain the required bandwidth of the data Flow, and sends a Port Stats request to exchange Port usage information. The Network_awareness module can perceive network topology information and link information, and monitor network data flow through Network_monitor. When the bandwidth required by the data stream is greater than the link bandwidth or the link bandwidth utilization is greater than the set threshold, the NSGAII_QoS module using multi-objective optimization genetic algorithm based QoS parameters to calculate the optimal routing. Then the rerouting configuration issue operation is performed by the Network_reconfiguration module.
### 3.2. Network topology and QoS parameter expression

The controller converts the collected network topology information and link information into an undirected graph $G(V, E)$ with $n$ nodes and $m$ connections. Where $V$ is the controlled switch and $E$ is the detected link, and it satisfies $|V| = n$, $|E| = m$. For the candidate path, can be expressed as $(X, List, Y)$, where $X$ represents the source host, $Y$ represents the destination host, and $List$ represents the set of links on the path from the source host to the destination host, $List$ is expressed as $List = \{l_1, l_2, ..., l_f\} (1 \leq f \leq m)$. There are QoS parameters on any link that can be used for multi-objective optimization of routing algorithms, such as link total bandwidth $b_{l_i}$, link residual bandwidth $d_{l_i}$, link bandwidth utilization $u_{l_i} = \frac{b_{l_i} - d_{l_i}}{b_{l_i}}$, link delay $d_{l_i}$, and link packet loss rate $p_{l_i}$. According to the QoS parameters on each link, the total QoS parameters on link set $List$ can be obtained. In these QoS parameters, The remaining bandwidth of the path is expressed as $A_{List} = \min \{a_{l_i} | l_i \in List\}$, the total bandwidth utilization of the path is expressed as $U_{List} = \max \{u_{l_i} | l_i \in List\}$, the total path delay is expressed as $D_{List} = \sum_{l_i \in List} d_{l_i}$, and the total packet loss rate of the path is expressed as $P_{List} = \sum_{l_i \in List} p_{l_i}$.

### 3.3. NSGAII_QoS routing implementation

The controller checks the amount of bandwidth required for the data flow and monitors link bandwidth utilization through the network_monitor module. When the system meets the following conditions, the network_reconfiguration module of controller will be called the NSGAII algorithm to reroute.

1. $W_b \leq A_{List}$
2. $U_{List} \leq 0.9$, where 0.9 is the threshold of the link bandwidth utilization selected in this paper.

To facilitate the interpretation of the routing algorithm in this paper, the network topology diagram designed is shown in Figure 2, and the routing implementation steps are expressed as follows:

Step1: Initialize the population. Randomly generate the initial population of population size $N$, that is, generate $N$ paths from $X$ to $Y$ in Figure 2. For the convenience of description, each list is assumed as a chromosome in this paper and some links are shown below,
Step 2: In order to enable the routing algorithm to make routing decisions according to the link information, this paper determines three multi-objective functions from the path remaining bandwidth, path time delay and path packet loss rate, which are expressed as

$$Object1(List) = \frac{1}{A_{list}}$$

$$Object2(List) = D_{list}$$

$$Object3(List) = P_{list}$$

Then, non-dominant sorting is carried out for some of the initial populations given in Step 1, and the obtained classification of different levels are expressed as

$$Rank0 = \{List3, List4\}$$

$$Rank1 = \{List2\}$$

$$Rank2 = \{List1\}$$

$$Rank3 = \{List5\}$$

$$Rank4 = \{List6\}$$

Step 3: The population number of the parent generation was set as $M$, and then $K$ chromosomes were randomly selected from the order of small to large, until the new population size reached the set population number of the parent generation, where the population number of the parent generation was satisfied $K < M$. Then the chromosome fragments with common genes in the new parent population formed after the selection operation were replaced, e.g., Both chromosome $X - A - B - C - D - E - Y$ and chromosome $X - F - B - C - D - E - Y$ have the same gene $B, D$, if the intersection point is $B$, the new chromosomes after crossing are $X - A - B - C - D - E - Y$ and $X - F - B - G - D - E - Y$, respectively. In addition, the mutation operation can randomly change a gene in the chromosome, e.g., for a new chromosome: $X - A - H - D - Y$, this chromosome can be regarded as a mutation from the chromosome $X - A - B - C - D - E - Y$. 

![Network Topology Diagram](image-url)
Step 4: The children and the parent are merged to perform the non-dominant sort again, and the crowding degree of each individual in the non-dominant layer is calculated during this sort process. According to the non-dominant relationship and the crowding degree of individuals, the appropriate individuals were selected to form a new parent population to achieve the elite retention strategy. In the calculation of crowding degree, the crowding distance should be set and the objective function should be normalized. The Normalization operation chosen in this paper is Min-Max Normalization [12], and the operating formula is as follows:

\[
\text{ObjFun}_1^\Delta(\text{List}) = \frac{\text{ObjFun}_1(\text{List}) - \min\{\text{ObjFun}_1(P)\}}{\max\{\text{ObjFun}_1(P)\} - \min\{\text{ObjFun}_1(P)\}} \tag{9}
\]

\[
\text{ObjFun}_2^\Delta(\text{List}) = \frac{\text{ObjFun}_2(\text{List}) - \min\{\text{ObjFun}_2(P)\}}{\max\{\text{ObjFun}_2(P)\} - \min\{\text{ObjFun}_2(P)\}} \tag{10}
\]

\[
\text{ObjFun}_3^\Delta(\text{List}) = \frac{\text{ObjFun}_3(\text{List}) - \min\{\text{ObjFun}_3(P)\}}{\max\{\text{ObjFun}_3(P)\} - \min\{\text{ObjFun}_3(P)\}} \tag{11}
\]

Set the front boundary \(I[i]_{\text{dist}, \text{fr}}\) and the back boundary \(I[\text{num}]_{\text{dist}, \text{fr}}\) of the non-dominant sort to be infinite, where \(\text{num}\) is the total number in the non-dominant layer. Then, the crowding degree of chromosome \(i\) in the non-dominant sequence is calculated according to the direction of each objective function, and the formula is expressed as

\[
I[i]_{\text{distance}} = I[i]_{\text{distance}} + \frac{\text{ObjFun}_1^\Delta(i + 1) - \text{ObjFun}_1^\Delta(i - 1)}{\max\{\text{ObjFun}_1^\Delta\} - \min\{\text{ObjFun}_1^\Delta\}} \quad (\omega = 1,2,3) \tag{12}
\]

Firstly, the crowding distance of each target function direction is added to obtain the crowding distance of this chromosome. Then, in order of the size of the crowding distance, the larger the crowding distance, the better the fitness, and therefore the more offspring there are. Finally, \(M\) parent populations are selected for a new genetic algorithm until the maximum number of iterations is satisfied.

3.4. The Description of NSGAII_QoS Routing Algorithm

In order to satisfy the dynamic QoS routing algorithm in the SDN network, the Network_monitor module of NSGAII_QoS must perform network monitoring at regular interval. When it is detected that the link utilization is high in the network or the bandwidth required by the data flow is greater than the link bandwidth threshold in the existing path, the Network_reconfiguration module starts rerouting, which selects the better candidate path through the congestion calculation of the elite policy. The algorithm description is as follows:

---

Algorithm 1 NSGAII_QoS routing algorithm

Input: \(G(V,E)\)

Output: forward path

1. Initialize the topology and define the chromosomes
2. Generate the initial population P
3. Define the objective function
4. \(k=1\)
5. While \((t=kT)\) do
6. Fast non-dominant sort of P
7. Crossover and mutation produce daughter population Q
8. while (the number of iterations is less than the maximum) do
9. \( R = P \cup Q \)
10. Fast non-dominant sort of \( R \)
11. Crowding is calculated and sorted
12. Select, cross, and mutate according to crowding
13. end while
14. \( k = k + 1 \)
15. end while
16. return forward path

4. Experimental testing and analysis

4.1. Experimental environment
In order to verify the feasibility of using the genetic algorithm based on elite strategy to make routing decisions. In this paper, Ryu 4.32 [13], mininet 2.3.0d5 [14] and OpenvSwitch [15] were used to build the SDN simulation environment on the Ubuntu 16.04 system. The network topology diagram used in the experiment is shown in Figure 3, which includes 7 virtual switches (s1-s7) and 8 hosts (h1-h8). The initialized network link bandwidth is shown in the link information in the figure, the delay of initialization link is set to 1ms and the packet loss rate of initialization link is set to 0%. Firstly, the iperf tool [16] was used in mininet to set h1-h4 as the client and h5-h8 as the server. Then, the client sends different rates of data streams to the server, gradually increasing the rate of data streams according to the step size of 5Mb/s. Finally, iperf tool is used to record the link throughput change, average link utilization, jitter change of transmission delay and packet loss rate in the topology network.

![Figure 3. Experimental network topology](image)

4.2. Experimental analysis
In the above experimental environment, the routing algorithm in this paper is compared with the ECMP algorithm [17] and the shortest path algorithm. Do the same experiment 20 times for each algorithm and record its average.

As shown in Figure 4, the link throughput of the shortest path algorithm, ECMP algorithm and the proposed NSGAII_QoS algorithm are compared. When the traffic does not reach the maximum forwarding capacity of the port, the throughput increases with the increase of the sending rate. In particular, when the client sending rate is less than 20Mb/s, the throughput of the three algorithms is
roughly the same, however, when the client sending rate is more than 20Mb/s, the routing algorithm in this paper will sacrifice some bandwidth for the interaction between the controller and the switch, but it can effectively reduce network congestion and reduce network packet loss, so as to improve the link throughput. Experimental results show that the routing algorithm in this paper is significantly larger than the other two routing algorithms in link throughput.

The comparison results of average link utilization are shown in Figure 5. The link utilization of the shortest path algorithm is still small and does not change much when the sending rate of the client increases gradually. And the ECMP routing algorithm will allocate the traffic of the big data stream to the path with a small amount of remaining link bandwidth, which leads to a low average link utilization. The experimental results show that the link utilization of the proposed algorithm is the highest among the three algorithms.

![Figure 4. The change of link throughput](image4.png)

![Figure 5. The change of link utilization](image5.png)

![Figure 6. The change of jitter delay](image6.png)

![Figure 7. The change of packet loss](image7.png)

Figure 6 shows the increase in the average delay of the three routing algorithms based on the original link delay of 1ms. Because the shortest path algorithm and ECMP algorithm do not consider the relationship between the bandwidth required by the data stream and the remaining bandwidth of the link, the forwarding delay changes greatly when the traffic transmission rate is large, and this increases the probability of network congestion. The experimental results show that the delay increment of the proposed routing algorithm is significantly smaller than that of the other two routing algorithms.

Figure 7 shows the network packet loss rates of the three algorithms at different transmission rates. The routing strategy based on the shortest path algorithm has a small network packet loss rate when the traffic data is small, but when the traffic data is large, if all the traffic is forwarded along the shortest path, it will easily lead to the shortage of link resources and network congestion, so the
network packet loss rate increases rapidly. For the ECMP algorithm, the traffic is evenly distributed among the existing path resources according to the number of traffic in the network at the same time. When the client sending rate is less than 20Mb/s, the network packet loss rate is small, but when the sending rate is more than 20Mb/s, the network packet loss rate also increases linearly. The experimental results show that the proposed algorithm in this paper can achieve a relatively low network packet loss rate.

5. Conclusion
Nowadays, how to guarantee the service quality of SDN has always been an important research topic. Since the original routing strategy is based on the shortest forwarding path of hops, this routing method is not flexible enough in the case of large data traffic emergencies, and it is easy to cause link congestion, which seriously affects the network service quality assurance. In this paper, according to the structure of SDN and the modular function of RYU controller, the NSGAII_QoS routing module is added in the RYU controller to achieve a better network performance. The module first obtains QoS parameters on the link by monitoring the underlying network, then combined with the bandwidth demand of the new data stream, a forwarding path which can satisfy the bandwidth demand and guarantee low transmission delay and network packet loss rate is found for the data stream through the multi-objective function optimization model, so as to alleviate the congestion in the network. The future work of this paper is to study the multi-objective algorithm to make routing decisions for different traffic flows, so as to further improve QoS assurance in SDN.

References
[1] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008) OpenFlow: enabling innovation in campus networks. J. ACM SIGCOMM Computer Communication Review., 38: 69-74.
[2] Masoudi, R., Ghaffari, A. (2016) Software defined networks: A survey. J. Journal of Network and computer Applications., 67:1-25.
[3] Hong, C. Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., &Wattenhofer, R. (2013, August) Achieving high utilization with software-driven WAN. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM. pp.15-26.
[4] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., ... & Zolla, J. (2013) B4: Experience with a globally-deployed software defined WAN. J. ACM SIGCOMM Computer Communication Review., 43:3-14.
[5] Yu, Y. S., & Ke, C. H. (2018) Genetic algorithm-based routing method for enhanced video delivery over software defined networks. J. International Journal of Communication Systems., 31:e3391.
[6] Dutra, D. L. C., Bagaa, M., Taleb, T., & Samdanis, K. (2017, December) Ensuring end-to-end QoS based on multi-paths routing using SDN technology. In GLOBECOM 2017-2017 IEEE Global Communications Conference. pp. 1-6.
[7] Zuo, Q.Y., Chen, M., Zhao, G.S., Xing, C.Y., Zhang, G.M., &Jiang, P.C. (2013) Research on OpenFlow-Based SDN Technologies. J. Journal of Software., 24:1078-1097.
[8] Li, B.K., Zhu, L., Ma, D., Hu, Y., Wang, G.Q., Jing, C.K. (2017) Routing Mechanism Based on Business Differentiating in Software Defined Network. J. Computer Science., 44:118-122.
[9] Zhou, F.J., Zhang, K.L., Wang, G.Q., Zhuang, L. (2019) Research of Adaptive SDN Routing Algorithm Based on Genetic Mechanism. J. Computer Engineering and Applications., 55:86-91+186.
[10] Peng, D.Q., Lai, X.W., Liu, Y.L. (2018) Multi-path Routing Algorithm for Fat-tree Data Center Network Based on SDN. J. Computer Engineering., 44:41-45+65.
[11] Jin, L., Shu, Y.A. (2019) Research on load balancing of elephant flow based on SDN in data center network. J. Application Research of Computers., 36:203-205.
[12] (2019) Min-Max Normalization [EB/OL]. http://webdataanalysis.net/data-analysis-method/data-normalization.
[13] (2019) Ryu [EB/OL]. http://sdnhub.org/tutorials/ryu
[14] (2019) Mininet [EB/OL]. http://mininet.org
[15] (2019) Openvswitch [EB/OL]. http://openvswitch.org
[16] (2019) Iperf[EB/OL]. https://iperf.fr
[17] Nong, H.W., Huang, C.H., Huang, X.P. (2016) SDN-based Multipath Routing Algorithm for Fat-tree Data Center Networks. J. Computer Science., 43:32-34+76.