Traffic Engineering Model and Algorithm Based on Bi-level Programming in Hybrid Software Defined Network

ZHE KANG
School of Management science and Engineering,
Shandong Normal University,
Jinan, China
MINGCHUN ZHENG*
School of Management science and Engineering,
Shandong Normal University,
Jinan, China
zhmc163@163.com

Abstract: -With the rapid development of Internet, the complexity of the network and control difficulty has increased dramatically. For the problem of fossilization network is more and more prominent, SDN (software defined network) arises at the historic moment. Due to SDN controller centralized control and the characteristics of shunt on SDN network equipment, this paper puts forward that part of nodes in the traditional network are changed to SDN nodes. In a hybrid SDN scenario, a new bi-level programming model in view of the virtual network by SDN nodes is proposed for traffic engineering algorithm. Using BLABG algorithm based on sensitivity analysis is to solve the model and to obtain the stability of the equilibrium solution. The traffic engineering model algorithm is more suitable for the network rush hour. It has good adaptability and can adjust the flow in the entire network to load balancing, and significantly reduces the delay effect of the underlying hybrid network and virtual network. Simulation results show that the proposed hybrid network traffic engineering algorithm is superior to the traditional network model.

Key-Words: - Hybrid SDN; Traffic Engineering; Bi-level Programming; Virtual Network

1 Introduction

SDN network is a new mode of network architecture. This network separates the network logic control from data forwarding, sets controller to centralized control over the entire network in the control plane and simplifies the complexity of the management of the router. The underlying device is only responsible for forwarding data. The network model has features as follow:[1] programmability, open the bottom switch programming interface, the user can according to their own requirements in terms of programming; Centralized control, in the control plane to provide visualization of the entire network information; Openness, data platform to the controller with a standard interface for programming the data platform and information gathering on the network state; Multistage flow table makes the flow more flexible and effective management.

More web applications can rapid deployment based on the characteristics of openness of SDN. Centralized network status information to traffic engineering research has injected new vitality. Due to the mastery of the entire network dynamic information, controlling flow on the decision-making level becomes more simple and effective.

At present, the industry and academia abroad have the application and research of SDN traffic engineering. In industry, Google’s B4 network[2] uses centralized traffic engineering. Through computing resource requirements, the network measures carrying capacity network to achieve multiplex forward tunnel and the dynamic weight allocation of bandwidth and uses the ECMP algorithm based on hash to achieve load balancing. Microsoft's SWAN[3], two different sharing strategies as follow: classifying flow and belonged to different priorities; the same priority flow follow the principle of maximum - minimum fair, so as to realize the network running efficiently. Bell Labs has launched a routing optimization control algorithm[4], using FPTAS (Fully Polynomial Time Approximation Scheme) to solve the problem of optimization of SDN controller, also is this link utilization rate of maximum-minimum problem. Academia's research mainly tends to SDN hardware implementation aspects of traffic engineering, the existing load balancing based on Open Flow equipment, such as: plug-in - n - serve, aster 'x, based on the load balancer Open Flow etc [1].

The above SDN network traffic engineering is deployed in the data centre network. Few scholars study at home and abroad the Internet traffic engineering on the backbone network in which SDN nodes is deployed, because the traditional backbone network not only large and having various complex protocols together. Based on this situation, ensure its effective operation, in recent years, studies have proposed part SDN nodes will be deployed to the network formed in the hybrid SDN network in order to improve the backbone. The network architecture can be either as a transitional phase of IP network to SDN, also can regard as the traditional network environment improvement. Stefano Vissicchio puts forward integrated hybrid model[5]. The model considers that SDN is responsible for all the web services, and using the traditional protocol as a node FIB (Forwarding Information Base) of the interface. It will put carefully chosen path into the routing system, and adjust the protocol parameters to control the forwarding path. In fact, each node of the FIB maintenance by the SDN, and controlled by SDN controller.

In this paper, based on the integration of the hybrid model, hybrid SDN network architecture is put forward. We need select some nodes into SDN nodes and consist of SDN virtual network. Through the collection of non SDN and SDN nodes state, the controller adopts bi-level programming model, considering the whole network efficiency, to implement traffic engineering on hybrid network. The architecture integrates the advantages of centralized and distributed network. In order to solve the contradiction of upper virtual network and underlying network, based on a Stackelberg game model, the upper SDN virtual network as a leader is centralized controlled by the controller, and aimed at the optimal system performance; according to upper virtual decision the lower hybrid SDN as follower makes advantage of the characteristics of distributed pursuing yourself utility maximum to make corresponding flow control decision. Finally coordinate with each other, system reaches stable state.

2 Related Knowledge

2.1 System Description

In the proposed hybrid SDN network, SDN nodes belong to a subset of the entire network nodes. There is a central controller used to control the SDN nodes, which receives the traditional networking protocol link information with the entire network routing logic, and calculates flexibly the routing to optimize routing table in SDN nodes. During the routing control in SDN nodes, controller has considered non-SDN nodes status and the current traffic demand model, so as to realize the flow split and path optimization. Because of the characteristic of the SDN centralized control, we suppose selected nodes in hybrid architecture as SDN nodes which consist of virtual network. As shown in Fig.1, the virtual network topology view exists in a centre controller. According to the upper user demand for virtual network traffic, the controller controls centrally SDN nodes to complete the upper network routing and traffic engineering. SDN node is a kind of hybrid router, includes both traditional network protocol stack, and can support the centralized traffic control of the controller. In addition to the instruction to forward data packets according to the controller, SDN node can also divide flow into different paths. In the underlying hybrid network non-SDN nodes support traditional OSPF protocol. They calculated in accordance with the principle of shortest path first routing and update the routing table. This hybrid network eventually improves the current traditional network performance.

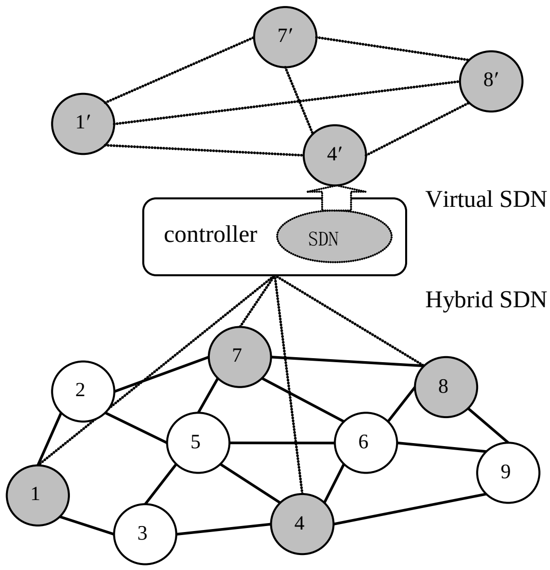

Fig. 1 SDN virtual network on hybrid network

Specific problem with reference to the literature[6] is as follows: we are given undirected network graphs $G = (V, A)$ (G denotes network) and $G' = (V', A')$ (G' denotes virtual network in which SDN nodes hand been selected). In SDN virtual network, a virtual link (i', j') connects adjacent SDN nodes and corresponds to a path from a SDN node i to the other one j in underlying hybrid network.

Virtual path on the SDN network is composed of virtual links between the SDN nodes. As shown in Fig 1, nodes 1, 4, 7, 8 as SDN nodes compose of the upper SDN virtual network, corresponding to 1′, 4′, 7′, 8′ in virtual network. Virtual link (1′, 7′) and (7′, 8′) compose a virtual path form 1′ to 8′.

In the underlying hybrid network, we use $a = (i, j)$ to denote a real link, y_a to denote flow on the link a, C_a to denote the biggest available bandwidth on a. q_{rs} represents traffic demand from node r to node s, which includes q_{rs1} (flow on SDN virtual network) and q_{rs2} (flow on underlying network) , $q_{rs} = q_{rs1} + q_{rs2}$. K_{rs} denotes the set of all paths in *O-D* pairs (r-s). f_k^{rs} denotes the flow between r and s on the first k ($k \in K_{rs}$) path. $\delta_{a,k}^{rs}$ indicates whether the link a on the first k path between r and s. If so, $\delta_{a,k}^{rs}$ is 1, otherwise 0.

$$y_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} \tag{1}$$

In hybrid SDN network, the controller has the routing logic information of the whole network, generates virtual network optimization routing according to the user's demand, and implements traffic control and network optimization on SDN nodes by centralized control. In routing optimization algorithms of SDN virtual network, $q_{r's'}$ represents traffic demand on virtual network from SDN node r' to SDN node s'. $K_{r's'}$ denotes a set of SDN virtual paths from r' to s' on virtual network. As shown in Fig 1, there are three virtual paths from 1′ to 8′. $K_{1'8'} = \{1' \to 7' \to 8', 1' \to 8', 1' \to 4' \to 8'\}$. k'($k' \in K_{r's'}$) denotes only a virtual path through SDN nodes. a' denotes a virtual link among two SDN nodes. $f_{k'}^{r's'}$ denotes the flow on the path k'. $C_{a'}$ denotes the bandwidth on the virtual link a'. $x_{a'}$ denotes the load of virtual link a'.

$$x_{a'} = \sum_{r'} \sum_{s'} \sum_{k'} f_{k'}^{r's'} \delta_{a',k'}^{r's'} \tag{2}$$

2.2 Optimization Model of Traffic Engineering

Within traditional AS of underlying network, ISP needs to ensure network load balance in order to provide more and better services. Traffic engineering as the tool of ISP network load balance can be implemented by using MPLS[7] to segment traffic between two nodes, through multiple preconfigured channels to achieve flow transmission between two nodes. Meanwhile use the optimization of OSPF[8-9] weight method to realize traffic engineering. In hybrid network with SDN nodes, we use the controller[10-13] centralized control to change the routing setting, making traffic load balance between SDN nodes. The underlying devices provide flow management[11, 14] function in order to realize flow division control.

The goal that traffic engineering pursuits is to balance the entire network load. There are two kinds of specific metrics, one is the pursuit of minimize the maximum load all link in entire network. The other one is the pursuit of minimum average delay in entire network. Due to the total delay of entire network can well reflect the entire network average delay, in this paper we use directly to minimize the total delay as traffic engineering programming objectives. Because the goals of two metrics are different, so the utility function is also different. We are given their formal description methods respectively.

Minimize the maximum link load. [15] regards the maximum link load as a metric of entire network flow balance. The flow of each link in the underlying network is given by (1), so the utility function for traffic engineering is $TE = \max_{a \in E} \frac{y_a}{C_a}$.

Minimize average delay of entire network. [16] takes advantage of the total delay of entire network as a way to reflect the average network delay. We assume that in a certain rush hour, the underlying switch equipment always batch packets. The time delay on a link is divided into two parts. The part is the propagation delay. Due to a short propagation delay, we do not consider the propagation delay in order to simplify the model. The other part is the queuing delay. Impedance that packets encounter in the process of the transmission is considered as delay. Assuming that only related to load with time delay, so the delay utility function for each link

$$t_a(y_a) = t_a^0 \cdot \left(1 + \alpha \left(\frac{y_a}{C_a}\right)^\beta\right) \tag{3}$$

y_a denotes the load of link a. t_a^0 denotes the basic processing time of the underlying switch equipment. α, β denote parameters.

To minimize the average delay of the entire network, we can reduce the effectiveness of each link to realize the entire network optimization. For each *O-D* pair when cannot find another path to make the average delay of entire network minimum, system can achieve the optimal equilibrium state.

2.3 Bi-level programming

Because the decision-making problem of the large system needs to consider the influence of various factors, its decision-making target involves the interests of the characters, and these goals and interests have mutual connection and mutual conflict in many cases, so it is necessary to adopt double decision method. Bi-level programming has considered decision-making targets of two decision-makers and the interaction between policy makers[17].

Upper model (U1)
$$\min_{x} Z(x,y) \quad s.t.\ G(x,y) \le 0$$

$y = y(x)$ by the lower level programming model

Lower model (L1)
$$\min_{y} z(x,y) \quad s.t.\ g(x,y) \le 0$$

Z is the objective function of the upper level programming. x denote decision variables for the upper programming. G is the constraint of variables. z is the objective function for lower level programming. y denotes the decision variables of lower programming. g is a constraint of variable y. The top decision makers influence for lower level decision makers by decision variable x, and interact with the lower decision makers. The lower decision variables is a function of the upper decision variables x. $y = y(x)$. The function is generally called reflect function.

3 The Traffic Engineering Model Based on Bi-level Programming

There are the SDN virtual network and traditional network in the proposed hybrid SDN network. We expect by SDN virtual network to improve the present situation of network and the overall network performance. In traffic engineering optimization problem, we consider using two layer networks to make decisions together. The upper virtual network is controlled by the controller in order to obtain minimum of overall system time delay, and the lower distributed network has an expectation for each *O-D* pair its delay in optimal. For two layer networks' relation of decisions and conflicting goals, the stackelberg game model is used to describe the relations. Assuming that the upper virtual network is the leader, the decision variable is X. The lower network for the follower, the corresponding decision variables is Y. The upper virtual network as a leader implements virtual network load balancing by the controller, and routing optimization model is established. The lower network as a follower, on the basis of flow distribution in the upper virtual link, decides to own control variables. Finally through limited iteration between the lower level network and the upper one, a balanced system is achieved.

Upper model

Because of the characteristic of the SDN network, controller controls the flow in order to obtain the minimum of total delay according to the information of entire network. So we can adopt the second principle of Wardrop to describe the model of system optimization (*SO*). Usually use shortest total travel time and generalized cost minimum to measure system performance, assumed that each *O-D* pair of virtual network on demand is fixed, the upper virtual network optimization model:

$$\min Z = \sum_{a'} x_{a'} t_{a'}(y_a(x_{a'}), x_{a'}) \qquad (4,5,6,7)$$

$$s.t. \begin{cases} \sum_{k'} f_{k'}^{r's'} = q_{r's'} & \forall r' \in R', s' \in S' \\ f_{k'}^{r's'} \ge 0 & \forall r' \in R', s' \in S', k' \in K'_{r's'} \\ x_{a'} = \sum_{r'}\sum_{s'}\sum_{k'} f_{a'}^{r's'} \delta_{a',k'}^{r's'} & \forall a' \in A' \end{cases}$$

R' is a set of starting SDN nodes that generates traffic demand. S' denotes a set of ending SDN nodes that attracts traffic. $r' \in R'$ denotes a starting SDN node. $s' \in S'$ denotes an ending SDN node.

Lower model

When the upper network optimizes network the virtual network need to consider the traffic distribution state of lower network. The underlying network as a follower according to upper network decisions optimizes the lower flow distribution, and the goal is to make each *O - D* pair on demand to minimize the time delay utility. We can describe the goal by the first principle (*UE*) of Wardrop. Usually use personal shortest travel time and generalized travel cost minimum to measure, the underlying network optimization model:

$$\min Z = \sum_{a} \int_0^{y_a} t_a(\omega) d\omega \qquad (8,9,10,11)$$

$$s.t. \begin{cases} \sum_{k} f_{k}^{rs} = q_{rs} & \forall r \in R, s \in S \\ f_{k}^{rs} \ge 0 & \forall r \in R, s \in S, k \in K_{rs} \\ y_a = \sum_{r}\sum_{s}\sum_{k} f_{k}^{rs} \delta_{ak}^{rs} & \forall a \in A \end{cases}$$

R is a set of starting nodes that generates traffic demand. S denotes a set of ending nodes that attracts traffic. $r \in R$ denotes a starting node. $s \in S$ denotes an ending node.

4 Algorithm Based on Sensitivity Analysis

In general, solving the bi-level programming problem is very complex. One of the reasons is that the bi-level programming problem is a NP - hard problem. There is no polynomial algorithm, often can only find the local optimal solution rather than the global optimal solution. Even if some kind of programming has an accurate algorithm, obviously is not suitable for large-scale network problems. So it usually adopts heuristic algorithm such as BLABG algorithm based on sensitivity analysis[18], BLABD algorithm based on differential calculation, etc. In this paper, we use the algorithm based on sensitivity analysis to solve the bi-level programming problem.

In this problem assumes that the time delay function (impedance) is strictly increasing and is only function of the link traffic. Thus the balance link traffic is a continuous function of SDN virtual link flow changes. In order to find out the approximate linear relationship between the balance link traffic flow and the virtual link, using sensitivity analysis obtain the derivative of balance link flow with respect to the adjusted value of virtual link flow.

The basic idea of this algorithm is as follows: set $x*$ to the initial value of SDN virtual link, y denotes the corresponding balance link traffic (calculated from the lower level problem).

$$y_a(x) \approx y_a(x*) + \sum_{i \in A} [\frac{\partial y_a(x)}{\partial x_i}]_{x=x'} (x_i - x'_i) \quad (12)$$

When (12) is put into to the upper objective function, the upper problem becomes a variable nonlinear optimization problem over virtual link traffic, which can use the existing method to solve. Find out the optimal solution for the top problem (that is, the new virtual network routing), to solve the problem of the lower again, a new balance link traffic on hybrid network may be made. Repeating the basic ideas, and can get a new set of SDN virtual network routing. Repeat calculation, and finally algorithm may converge to the optimal solution of bi-level programming model of the original. Specific algorithm steps are as follows:

Step1: setting initial solution of SDN virtual link traffic to x^0, the number of iterations is $k = 0$;

Step2: for a given x^k, solving lower level problem, get equilibrium link traffic y^k on hybrid network;

Step3: sensitivity analysis method is used to calculate the derivative of balance link traffic y^k with respect to virtual link traffic;

Step4: put (12) into the upper objective function, to solve the upper problem, get a new set of virtual link traffic x^{k+1};

Step5: if $\max\left|x^{k+1} - x^k\right| \le \sigma$, algorithm is stopped, σ is the iteration precision; otherwise, $k = k + 1$, turn to step2.

Corresponding to different SDN routing schemes, it can get different y that is reflected in the routing of hybrid network according to the underlying distributed principle selected routing. Thus we get different control scheme of system flow, and can choose the optimal solution by comparison.

5 Simulation Experiment

5.1 The Setup of Experimental and Evaluation Standard

At present, the virtual network routing and the underlying network traffic engineering still choose NS2 simulation environment. The traditional overlay of traffic engineering has the algorithm based on leader preferred strategy (PS) and traditional non-cooperative game model (NonCo), etc. Here we contrast BLABG algorithm proposed in this paper with the traditional algorithm through the experiment. In order to verify the stability and effectiveness of the proposed algorithm, our algorithm runs on Matlab platform. First set the underlying hybrid network topology for 20 nodes and 80 links. Second randomly select 7 SDN nodes from 20 nodes to form SDN virtual network. The hybrid network link bandwidth obeys uniform distribution of 30-100. Assume that the traffic demand is uploaded between any two points, which are on the underlying hybrid network and the SDN virtual network, according to a given traffic matrix M, elements in the matrix M denote the traffic demand between SDN nodes that is the sum of virtual network traffic and hybrid network traffic. Assuming that the proportion of the flow of virtual network that accounts for hybrid network is 0.4. In order to compare the effectiveness of time delay utility decrease which SDN nodes bring in peaks and troughs hours, we set M, 2M, 5M, 10M to compare their performance respectively.

In order to compare with traditional algorithm, we use the following three evaluation standards for each algorithm to evaluate its performance. First is the hybrid network performance. It can use the average delay of entire network or the maximum link load as evaluation standard of the underlying network performance. For algorithm performance unified, selecting entire network average delay is as evaluation standard. Second is SDN virtual network performance. This paper uses total delay on the virtual network to reflect the virtual network performance. The last one is link load distribution. It refers that when the algorithm terminates and system is stable, whether the flow on a link distribution is uniform or not. So this standard reflects the algorithm has an optimize performance of network traffic control.

5.2 The Simulation Results and Comparison

In order to compare the impact of the decision of the three kinds of algorithms for the different demand of the network, we adjust to the traffic matrix to compare the end result of three algorithms. Fig.2 shows optimal solution of three objective functions respectively in the M, 2M, 5M, 10M cases. From the vertical and horizontal axis, on hybrid network and SDN virtual network, the total delay with the increase of network demand is gradually increasing. And in less demand for traffic, three kinds of algorithm get the similar total delay. With the increase of demand for network, BLABG algorithm solution is better than that of PS and NonCo gradually. And this advantage with the increase of demand is more and more obvious. It is verified that BLABG algorithm in network peak can further highlight the optimality.

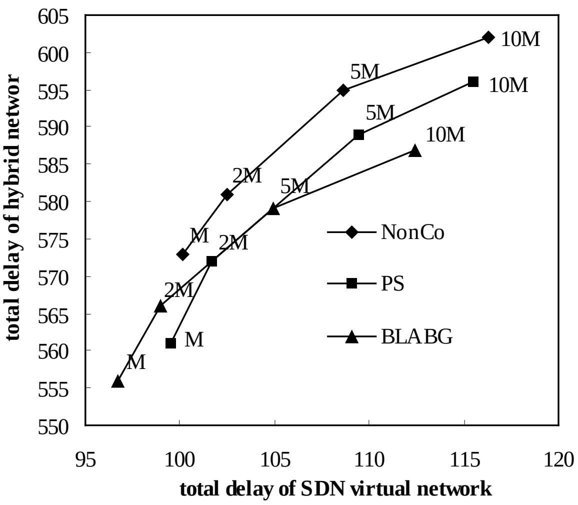

Fig.2 Total delay on hybrid network and SDN virtual network

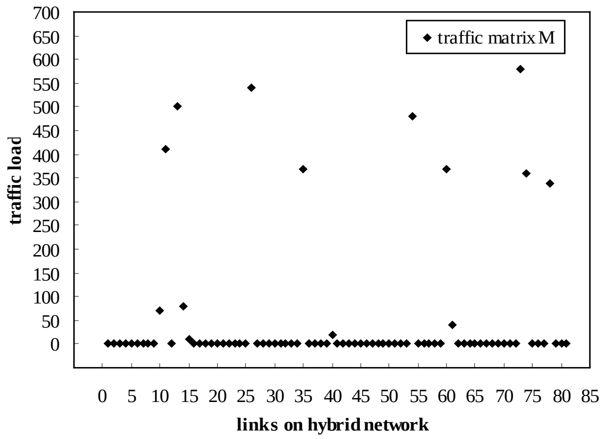

Fig.3 Network traffic distribution (matrix M)

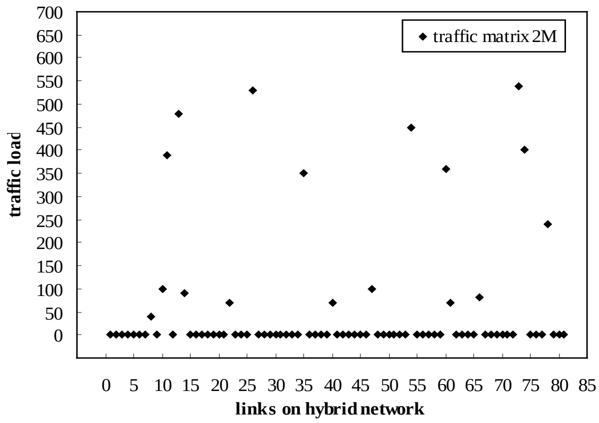

Fig.4 Network traffic distribution (matrix 2M)

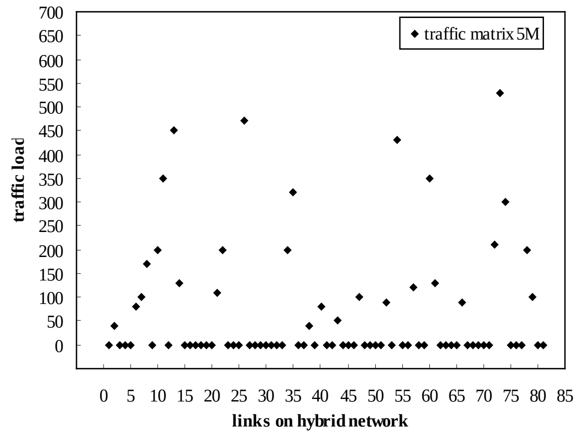

Fig.5 Network traffic distribution (matrix 5M)

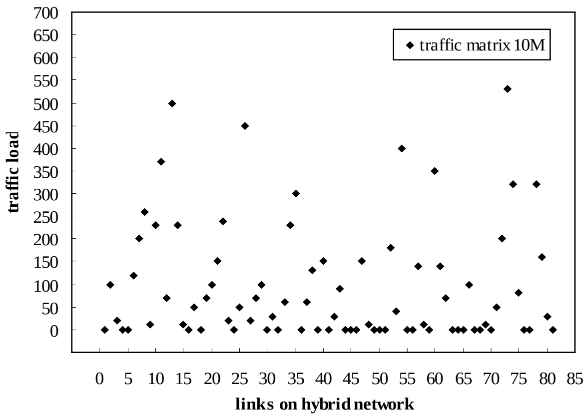

Fig.6 Network traffic distribution (matrix 10M)

As can be seen from the above four figures (Fig.3, Fig.4, Fig.5, Fig.6), when the network demand matrix is M, only 14 links have traffic. And with the increase of demand, quantity of active link is increasing. When demand matrix is the 10M, almost all of the links are assigned to the traffic. As you can see, BLABG algorithm proposed in this article is self-adapting and can allocate traffic to the link according to the size of the demand of network. Load balancing degree on the network can be adaptive to the demand. Ultimately hybrid network system achieves stable and has the high efficiency.

6 Conclusion

Based on hybrid SDN network traffic engineering problem is studied. This paper applies stackelberg game theory to SDN network and build SDN virtual network, thus form two decision-making level with underlying network. Combining with the bi-level programming model makes two layers of network to control traffic. The upper goal is to achieve the optimal system. The lower goal is to make the user optimal. BLABG algorithm based on sensitivity analysis is to solve the bi-level programming model. Finally we get the stackelberg stable equilibrium solution and make the system in balance. Based on the above model and algorithm, the simulation experiment data show that the hybrid network traffic engineering model based on bi-level programming is better than traditional model algorithm. At the same time, this model also exist some shortage. In practice traffic demand on the network is stochastic and dynamic, virtual network is generated through optimizing. In this paper, we assume that virtual network has existed in the hybrid network, and $O-D$ pair of traffic demand is fixed. For the next step we should research to be more close to reality, relax more constraint conditions and gradually deep into actual SDN network.

Acknowledgements

This work is supported by the Natural Science Foundation of China (NO.61402266), Natural Science Foundation of Shandong Province, China (ZR2012MF013), Social Science Foundation of China (14BTQ049). We thank our shepherd Ming-chun Zheng and the anonymous reviewers for comments.

References:

[1] Ian F. Akyildiz, Ahyoung Lee, Pu Wang. *A roadmap for traffic engineering in SDN-OpenFlow networks [J]*. Computer Networks, 2014, 71:1-30.

[2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S.Venkata, J. Wanderer, J. Zhou, M. Zhu, et al. *B4: experience with a globally-deployed software defined wan[J]*. Proceedings of the ACM SIGCOMM Conference, SIGCOMM'13, August 2013, pp. 3–14.

[3] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, R.Wattenhofer. *Achieving high utilization with software-driven wan[J]*. Proceedings of the ACM SIGCOMM 2013, August 2013, pp. 15–26.

[4] S. Agarwal, M. Kodialam, T. Lakshman. *Traffic engineering in software defined networks[J]*. Proceedings of the 32nd IEEE International Conference on Computer Communications, INFOCOM'13, April 2013, pp. 2211–2219.

[5] Stefano Vissicchio, Laurent Vanbever, Olivier Bonaventure. *Opportunities and Research Challenges of Hybrid Software Defined Networks.* ACM Computer Communication, 2014, 44(2).

[6] Peng Wang, Junzhou Luo, Wei Li. *A cooperative game model for traffic engineering and overlay routing based on trustworthy and controllable network [J]*. Chinese Journal Computers, 9: 1663-1674 (2010). (In Chinese)

[7] Davie B, Rekhter Y. *MPLS: Technology and Applications.* San Francisco, CA: Academic Press, 2000.

[8] Fortz B, Thorup M. *Internet traffic engineering by optimizing OSPF weights.* Proceedings of the INFOCOM. TelAviv, Israel, 2000: 519-528.

[9] Shrimali G et al. *Cooperative inter-domain traffic engineering using Nash bargaining and*

decomposition. Proceedings of the INFOCOM. Anchorage, Alaska, USA, 2007: 330-338.

[10] A. Tootoonchian, Y. Ganjali. *Hyperflow: a distributed control plane for openflow*. Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, INM/WREN'10, April 2010, p. 3.

[11] M. Yu, J. Rexford, M.J. Freedman, J. Wang. *Scalable flow-based networking with difane.* ACM SIGCOMM Comp. Commun. Rev. 40(4) (2010) 351–362.

[12] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al. *Onix: a distributed control platform for large-scale production networks.* Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI'10, vol. 10, October 2010,pp. 1–6.

[13] Y. Hu, W. Wang, X. Gong, X. Que, S. Cheng. *Balanceflow: controller load balancing for openflow networks.* Proceedings of IEEE 2nd International Conference on Cloud Computing and Intelligent Systems, CCIS'12, vol. 2, October 2012, pp. 780–785.

[14] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S.Banerjee. *Devoflow: scaling flow management for high-performance networks.* ACM SIGCOMM Comp. Commun. Rev. 41(4) (2011) 254–265.

[15] Seetharaman S, Hilt V, Hofmann M, Ammar M. *Preemptive strategies to improve routing performance of native and overlay layers.* Proceedings of the IEEE INFOCOM. Anchorage, Alaska, USA, 2007: 463－471.

[16] Liu Y, Zhang H, Gong W, Towsley D. *On the interaction between overlay routing and traffic engineering.* Proceedings of the IEEE INFOCOM. Miami, FL, USA, 2005: 2543-2553.

[17] Ziyou Gao. *Bi-level programming models, approaches and applications in urban transportation network design problems [J]*. Journal of Transportation Systems Engineering and Information Technology, 2: 35-36 (2004). (In Chinese)

[18] Tobin R L, Friesz T L. *Sensitivity analysis for equilibrium network flow [J]*. Transportation Science, 1998, 22:242－250.