



Intelligent Dynamic Timeout for Efficient Flow Table Management in Software Defined Satellite Network

Shahid Jan¹(✉), Qing Guo¹, Min Jia¹, and Muhammad Kamran Khan²

¹ School of Electronics and Information Engineering,
Harbin Institute of Technology, Harbin, China

sjanwardag@yahoo.com,
{qguo, jiamin}@hit.edu.cn

² Department of Electronics, University of Peshawar, Peshawar, Pakistan
kamranmu@uop.edu.pk

Abstract. Software Defined Network (SDN) modify the architecture of traditional satellite network into Software Defined Satellite Network (SDSN) by decoupling its control and data planes. However, SDSN encounter several issues, such as satellite link handover and limited space of Ternary Content Addressable Memory (TCAM), which results into increasing the number of flow rule entries and flow drop. To solve these issues this paper presents a novel three-layer architecture of SDSN and propose Intelligent Dynamic Timeout (IDT) algorithm. The algorithm predicts dynamic timeout for the eviction of unused flow entries in order to reduce the size of flow table, drop flow rate and number of table miss packets. Simulation results show that the average size of flow table, drop flow rate and number of table miss packets are reduced by 39.55%, 11.2% and 10.18% respectively when comparing the performance of IDT with different static idle timeout values.

Keywords: Software Defined Satellite Network ·
Flow table management · Dynamic timeout · IDT

1 Introduction

Satellite communication play a vital role in information transmission and it has been widely used for military purposes, live television, weather broadcast and internet of things [1]. However, traditional satellite network led the resources underutilization and it does not guarantee the requirements of future [2]. Software Defined Network (SDN) is an emerging network paradigm, which modify the architecture of distributed network by decoupling its control and data planes [3]. Traditional satellite network has adopted the architecture of SDN and results into Software Defined Satellite Network (SDSN). SDSN is explained by many recent researches e.g. OpenSAN [4] divide the architecture of SDSN into three planes. Similarly, [5] also present the architecture of SDSN.

SDN switches install flow rules given by controller in Ternary Content Addressable Memory (TCAM) and a single SDN switch can accommodate 1500 flow rules, because TCAM are power hungry and more costly [6]. In SDSN frequent satellite link handovers occur in network topology, which create more flow table entries and results into flow drop [7]. Different timeouts, ranging from 5 s to 60 s, have been used to evict the unused flow entries from switch [8]. However, all these timeouts are fixed and do not consider the occupied space of the switch.

This paper uses dynamic timeout to evict the unused flow entries from switch and thus reduces the size of flow table, drop flow rate and number of table miss packets. The technical contributions of this paper are given below.

- It presents a novel three-layer architecture of SDSN and propose an algorithm of Intelligent Dynamic Timeout (IDT) for efficient flow table management.
- The proposed algorithm considers the key points of limited TCAM space at forwarding satellites and priority based classified traffic of satellite network to predict a dynamic idle timeout.
- Simulation results show that the average size of flow table, drop flow rate and number of table miss packets are reduced by 39.55%, 11.2% and 10.18% respectively when comparing the performance of IDT with the static idle timeout of different values.

The rest of this paper is organized as follows. Section 2 explains related work. Section 3 explains the proposed architecture of SDSN and algorithm. Section 4 explains prototype implementation. Section 5 explains the experimental results and performance analysis. Finally, Sect. 6 conclude this paper.

2 Related Work

Flow Table management in ground network is explained by many researches and we have divided it into three directions. The first direction explains the designing of local policies inside SDN switches to reduce the number of requests sent to the controller. This concept has been used by DevoFlow [6] to reduce the number of flow entries in TCAM of the switch and minimize the communication between switch and controller. Similarly, DIFANE [8] uses middle switches in data plane and install flow rules in these switches to keep the traffic flow in data plane.

Second direction is the use of wildcard rules to reduce the size of flow table in switches. Leng et al. [9] proposed the scheme of flow table reduction by splitting the large groups of addresses into smaller groups and rewrite smaller groups in flow table. H-SOFT [10] proposed optimization algorithms to reduce the high-dimensional and complex fields of flow table into low-dimensional and multiple flow tables.

Third direction is to assign an appropriate static or dynamic idle timeout values to flush out the unused flow entries. Zarek et al. [11] explains the assigning of proper static idle timeout to evict flow entries from the flow table. Zhu et al. [12] proposed a scheme to assign different timeout to the newly arrived flows according to their characteristics. Vishnoi et al. [13] present Smart Time which

combines adaptive timeout with proactive eviction of flow rules to calculate well-organized idle timeout and result in effective utilization of TCAM space. However, all these approaches are not appropriate for flow table management in SDSN, because it does not consider the satellite link handover, limited resources of satellite network and its priority based classified traffic.

3 SDSN Architecture and Algorithm

3.1 SDSN Architecture

The proposed SDSN architecture is based on the basic paradigm of SDN and comprises of three planes: Control Plane (GEO Satellites), Data Plane (LEO Satellites) and Management Plane (Ground Stations) as illustrated in Fig. 1.

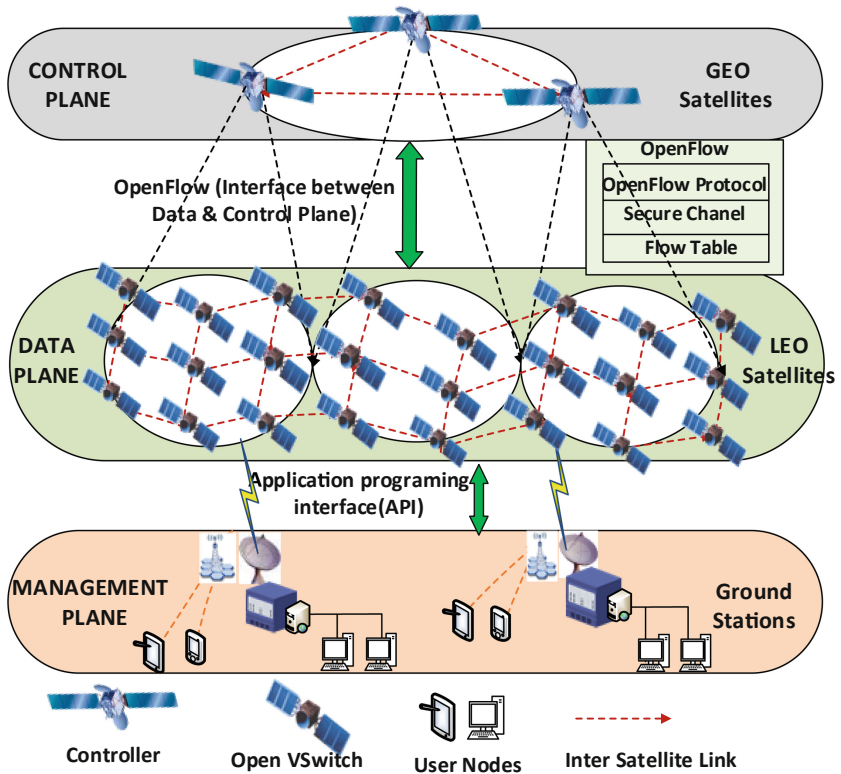


Fig. 1. Software Defined Satellite Network architecture

Control Plane: In this plane Geostationary Earth Orbit (GEO) satellites reside due to its wide coverage and stationary to the ground characteristics. This framework contains three GEO satellites for monitoring and control of data

plane. GEO satellites monitor the LEO satellites flow status, link status and share it with the management plane for satellite network view. It also receives the routing rules from data plane and translate it to OpenFlow rules.

Data Plane: This architecture consider Low Earth Orbit (LEO) satellites in data plane due to less distance from the ground. A constellation of 40 LEO satellites is enough for global seamless coverage. Each satellite has four intersatellite links: two bidirectional links with the satellites of same orbit and two links with the corresponding satellites of different orbits. Each LEO Satellite contains Open Vswitch to run flow table and perform the packets forwarding function.

Management Plane: The ability of satellite to process network routing algorithms, perform the task of network security and manage all other resources are limited, therefore management plane will process all these responsibilities. The view of whole network collected from GEO satellites by Ground Stations will make able the management plane to run network policies. These network policies will be distributed among LEO satellites through GEO satellites.

3.2 Algorithm

To achieve the goal of efficient flow table management in SDSN, this paper proposes IDT algorithm which consist of two different Modules as illustrated in Fig. 2. The Feedback Module sends tableStatus request to each switch after every second and records the last value of flow entries. The Timeout Prediction Module Predict timeout according to the occupied space of TCAM in switch. Table 1 list the definition of basic parameters used by this algorithm.

IDT algorithm consider two important ideas. Firstly, due to limited TCAM space in SDSN, IDT predict timeout for a flow entry based on the occupied

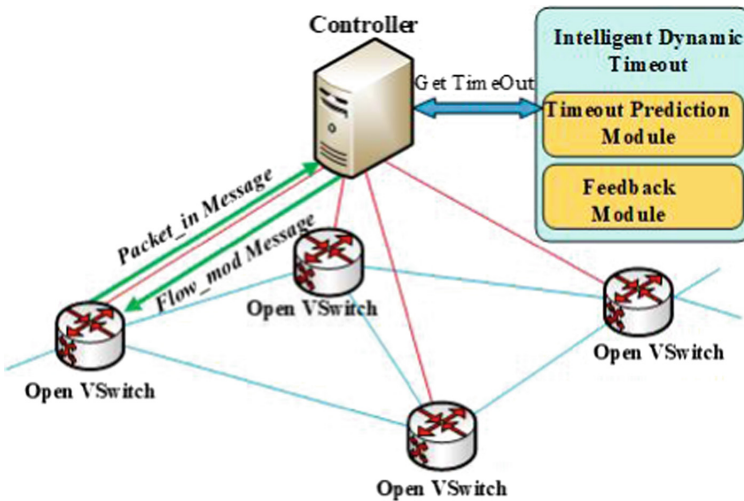


Fig. 2. Intelligent dynamic timeout modules description

Table 1. Definitions of basic parameters.

Parameters	Definitions
TCAM(peak)	Total TCAM space boundary limit
TCAM(max)	90% of the Total TCAM space boundary limit
TCAM(start)	80% of the Total TCAM space boundary limit
TCAM_Occup	Current TCAM occupied space by flow rule entries
T(idle)initial	Initial idle timeout
T(idle)ct	Initial idle timeout used for classified traffic
df1, df2, df3	Decreasing factor for traffic of group1, group 2 and group 3
T(idle)flow(x)	Idle timeout install for flow(x)
T(idle)IDT	The idle timeout of flow(x) by IDT

space of the switch and decreases the value of timeout according to increase in number of flow entries. Secondly, due to limited resources of SDSN it is difficult to allocate each user enough resources. Thus, we have classified users into three different groups based on their MAC/IP addresses. The timeout of these groups reduces in different proportion according to the decreasing factors.

Algorithm 1. Intelligent Dynamic Timeout Algorithm

Input: TCAM(peak), TCAM(max), TCAM(start), T(idle)ct, T(idle)initial, df1, df2, df3

Output: T(idle)IDT

For every Switch: Monitor the number of Flow entries and Record the last n value

```

1: for Packet_in message do
2:   Check TCAM_Occup:
3:   if TCAM_Occup < TCAM(start) then
4:      $K = (TCAM\_Occup - TCAM(start)) / TCAM(max)$ 
      $T(idle)flow(x) = T(idle)initial - (K \times T(idle)initial)$ 
5:   else if TCAM(start) <= TCAM Occup < TCAM(max) then
6:      $K = (TCAM\ Occup - TCAM(start)) / (TCAM(start) - TCAM(max))$ 
      $T(idle)flow = T(idle)initial - (K \times (T(idle)initial - 6))$ 
7:   else if TCAM_Occup >= TCAM(max) then
8:     if user  $\in$  group 1 then
9:        $T(idle)flow(x) = T(idle)ct \times df1$ 
10:    else if user  $\in$  group 2 then
11:       $T(idle)flow(x) = T(idle)ct \times df2$ 
12:    else if user  $\in$  group 3 then
13:       $T(idle)flow(x) = T(idle)ct \times df3$ 
14:    end if
15:  end if
16:   $T(idle)IDT = T(idle)flow(x)$ 
17: end for
18: return  $T(idle)IDT$ 

```

The execution process of IDT is divided into five Phases. Phase 1 accepts the input data. We consider $T(\text{idle})_{\text{initial}}$ as 12 s, $T(\text{idle})_{\text{ct}}$ as 10 s. $df1$, $df2$ and $df3$ are 0.7, 0.6 and 0.5 respectively. Phase 2 check flow table statistics every second and record the last value of flow table entries in a csv file. In Phase 3, the controller check TCAM occupied status of each switch. As $T(\text{idle})_{\text{initial}}$ is 12s, if there is no flow table entry $T(\text{idle})_{\text{IDT}}$ will be 24s. As the number of flow entries increases, it decreases the timeout linearly and when the TCAM Occupied space reaches to $\text{TCAM}(\text{start})$, timeout decreases to 12s. In phase 4, the idle timeout value decreases fast to avoid reaching the $\text{TCAM}(\text{peak})$. In Phase 5, timeout value decreases suddenly in correspondence with the decreasing factors, to ensure the traffic flow from high priority users and avoid flow drop. The time complexity of IDT algorithm is $O(n)$.

4 Implementation

To implement IDT algorithm, we used Mininet, Pox Controller and Open Vswitches to create a realistic virtual network. In addition, we developed 2-layer satellite constellation in Satellite Tool Kit (STK) with the parameters of its orbit as listed in Table 2 [14]. Figure 3 illustrate 3D view of the whole constellation.

Table 2. Parameters of GEO and LEO satellite orbits.

Orbit planes	Altitude in km	No of satellites	Inclination	True anomaly
GEO	36000	3	0°	0
LEO 1	887	8	86°	0
LEO 2	887	8	86°	333
LEO 3	887	8	86°	353
LEO 4	887	8	86°	328
LEO 5	887	8	94°	351

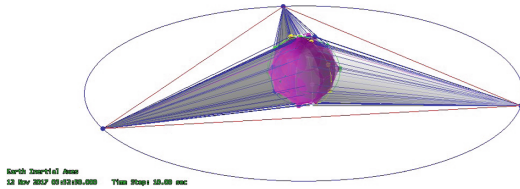


Fig. 3. 3D view of satellite constellation

The designed constellation has two ground stations at Sydney and Beijing. The purpose of satellite constellation is to measure satellite access to the ground

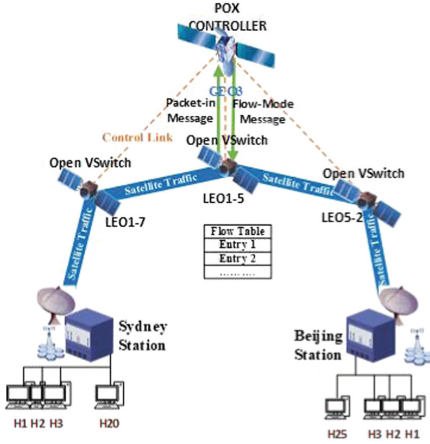


Fig. 4. Network topology

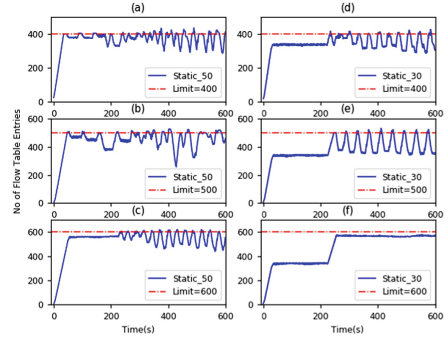


Fig. 5. Flow entries of static timeout 50 s and 30 s

stations which leads us to define a static network topology as illustrated in Fig. 4. This topology is fixed and use only three satellites of LEO1-7, LEO1-5 and LEO5-2 to transmit data from 20 hosts connected to Sydney station to the 25 hosts connected to Beijing station as illustrated in Fig. 4.

5 Experimental Results and Performance Analysis

5.1 Experimental Results

Four sets of experiments are created to estimate the performance of our proposed algorithm. Time of each experiment is 10 min (600 s). In each experiment random traffic flows from 20 hosts connected to Sydney station to the 25 hosts connected to Beijing station as illustrated in Fig. 4.

First experiment considers static idle timeout of 50 s. The results of this timeout strategy under TCAM space limit 400, 500 and 600 are shown in Fig. 5. (a), (b) and (c) respectively. In each case, the number of flow table entries cross the final limits, because the timeout value is larger and the unused flow entries reside in switch for more time due to which packets drop will occur. Second experiment considers static idle timeout of 30s. Simulation results of this timeout strategy is explained in Fig. 5. (d), (e) and (f) under the TCAM limit 400, 500 and 600 respectively. Figure 5(d) and (e) show that the number of flow table entries cross the final limit of TCAM due to maximum timeout which result into packets drop. But in Fig. 5(f) the maximum number of flow table entries do not cross the TCAM limit, because the TCAM space is enough to accommodate more flow entries. There will be no packets drop in this case.

Third experiment considers static idle timeout of 15 s. Results of this timeout strategy under the TCAM space limit of 400, 500 and 600 are shown in Fig. 6. (a),

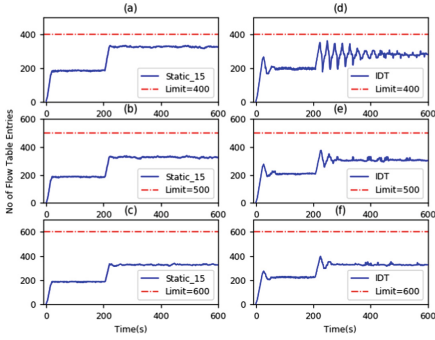


Fig. 6. Flow entries of static timeout 15 s and IDT

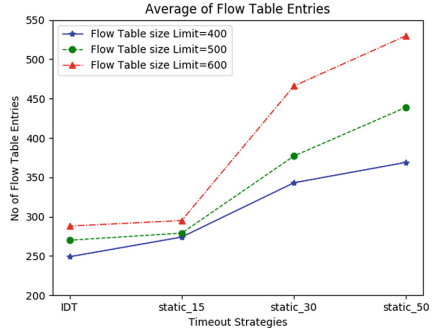


Fig. 7. Comparison of flow entries.

(b) and (c) respectively. In each case, number of flow table entries do not cross TCAM limit because the idle timeout is less and it evict the flow entries more quickly. In this case there will be no packets drop, but the number of table miss packets will increase. Fourth experiment considers dynamic idle timeout of IDT to evict the flow table entries from switch. The results of IDT under the TCAM space limit 400, 500 and 600 are shown in Fig. 6. (d), (e) and (f) respectively. In each case, flow table entries do not reach the final limit of TCAM space and there are no packets drop in IDT.

5.2 Performance Analysis

(a) Comparison of Flow Table Entries: The Comparison of average flow table entries of different timeout under the TCAM limit 400, 500 and 600 are shown in Fig. 7. The analysis shows that, Static_30 reduced the average number of flow table entries as compared to Static_50 by 7.04%, 14.12% and 12.07% under the TCAM limit 400,500 and 600 respectively. Similarly, Static_15 reduced the average number of flow table entries as compared to Static_30 by 20.11%, 25.99% and 36.69% under the TCAM limit of 400, 500 and 500 respectively. IDT reduced the average number of flow table entries as compared to Static_15 by 9.12%, 3.22% and 2.37% under the TCAM limit of 400, 500 and 600 respectively. Thus, using dynamic timeout of IDT instead of static idle timeout for evicting flow entries, can shrink the size of flow table dramatically.

(b) Drop Flow Rate: Figure 8 illustrates the drop flow rate of Static_50, Static_30, Static_15 and IDT. Here the drop flow is only due to the limited space of TCAM. Figure 8 shows that, the drop flow rate of static_50 is reduced 10.12%, 10.28% and 4.54% by static_30 under the TCAM limit of 400, 500 and 600 respectively. Similarly, drop flow rate of Static_30 is reduced 5.76%, 2.47% and 0.06% by Static_15 and IDT under the TCAM limit of 400, 500 and 600 respectively. The drop flow rate decreases when TCAM space limit increases and

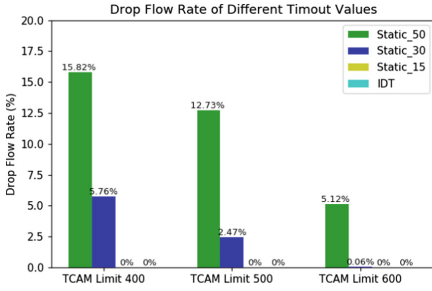


Fig. 8. Drop flow rate of different timeout.

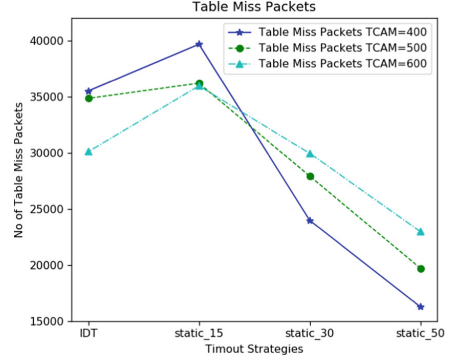


Fig. 9. Table miss packets of different timeout.

the value of timeout decreases. In our case as the idle timeout approaches to 15s drop flow rate turns to 0%. But using static timeout of less value triggers more table miss packets, which create additional overheads on controller. Thus, using dynamic timeout like IDT is more suitable in SDSN than static timeout of less value.

(c) **Table Miss Packets:** Figure 9 illustrates the total number of table miss packets, flows from three switches at LEO satellite nodes to one controller at GEO satellite node. When TCAM limit is 400, in Static_50 table miss packets are 32.1% less than Static_30. In Static_30 table miss packets are 39.58% less than Static_15. Similarly, in IDT table miss packets are 10.51% less than Static_15. When the TCAM limit is 500, in Static_50 table miss packets are 29.43% less than Static_30. In Static_30 table miss packets are 22.90% less than Static_15. In IDT table miss packets are 3.74% less than Static_15. When TCAM limit is 600, in Static_50 table miss packets are 23.32% less than Static_30. In Static_30 table miss packets are 16.71% less than Static_15 and table miss packets of IDT are 16.29% less than Static_15. Thus, using static timeout of more values reduces the number of table miss packets but increases packets drop while timeout of less value avoid packets drop but increases the number of table miss packets which is unacceptable in case of SDSN.

6 Conclusion

This paper has addressed issues of flow table management faced by SDSN. A scalable novel three-layer architecture of SDSN and IDT algorithm is proposed to solve these issues. The algorithm considers limited resources of satellite network and its priority based classified traffic to predict a dynamic idle timeout. In addition, a satellite constellation has been developed to evaluate the scalability

of the proposed architecture and achieve the real parameters of satellite network. Extensive Simulation experiments have been performed to compare the performance of IDT algorithm with static timeout. The performance analysis demonstrated that IDT algorithm can efficiently reduce the size of flow table, drop flow rate and number of table Miss packets.

References

1. Dexin, D., Zengwei, Z., Meimei, H.: A survey: the progress of routing technology in satellite communication networks. In: 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, pp. 286–291 (2011)
2. Li, T., Zhou, H., Luo, H., Yu, S.: SERvICE: a software defined framework for integrated space-terrestrial satellite communication. *IEEE Trans. Mob. Comput.* **17**(3), 703–716 (2018)
3. Kreutz, D., Ramos, F.M.V., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. *Proc. IEEE* **103**(1), 14–76 (2015)
4. Bao, J., Zhao, B., Yu, W., Feng, Z., Wu, C., Gong, Z.: OpenSAN: a software-defined satellite network architecture. In: ACM SIGCOMM Computer Communication Review, vol. 44, no. 4, pp. 347–348. ACM (2014)
5. Bertaux, L., et al.: Software defined networking and virtualization for broadband satellite networks. *IEEE Commun. Mag.* **53**(3), 54–60 (2015)
6. Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Banerjee, S.: DevoFlow: scaling flow management for high-performance networks. *ACM SIGCOMM Comput. Commun. Rev.* **41**(4), 254–265 (2011)
7. Li, T., Zhou, H., Luo, H., You, I., Xu, Q.: SAT-FLOW: multi-strategy flow table management for software defined satellite networks. *IEEE Access* **5**, 14952–14965 (2017)
8. Yu, M., Rexford, J., Freedman, M.J., Wang, J.: Scalable flow-based networking with DIFANE. *ACM SIGCOMM Comput. Commun. Rev.* **40**(4), 351–362 (2010)
9. Leng, B., Huang, L., Wang, X., Xu, H., Zhang, Y.: A mechanism for reducing flow tables in software defined network. In: 2015 IEEE International Conference on Communications (ICC), London, pp. 5302–5307 (2015)
10. Jingguo, G., Zhi, C., Yulei, W., Yuepeng, E.: H-SOFT: a heuristic storage space optimization algorithm for flow table of OpenFlow. In: 2014 14th January Concurrency and Computation, pp. 3497–3509. Wiley Online Library (2014)
11. Zarek, A., Ganjali, Y., Lie, D.: OpenFlow Timeouts Demystified. University of Toronto, Toronto, Ontario, Canada (2012)
12. Zhu, H., Fan, H., Luo, X., Jin, Y.: Intelligent timeout master: dynamic timeout for SDN-based data centers. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 734–737. IEEE (2015)
13. Vishnoi, A., Poddar, R., Mann, V., Bhattacharya, S.: Effective switch memory management in OpenFlow networks. In: Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, pp. 177–188. ACM (2014)
14. Qu, Z., Zhang, G., Cao, H., Xie, J.: LEO satellite constellation for internet of things. *IEEE Access*, **5**, 18391–18401 (2017)