Smooth Transition Process of Sleep Mechanism Based on SDN Framework

Xu Lei[1,2], Gu Jinguan[1,2], He Heng[1,2]

1.School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan
430065, China;
2.Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System,Wuhan University of Science and Technology, Wuhan 430065, China
[1]rockstore@163.com, [2]heheng@wust.edu.cn

Keywords: Sleep mechanism, SDN, Smooth transition.

Abstract. Green network uses sleep mechanism for energy-saving design, achieving sleep mechanism needs to consider that the mechanism may cause data loss, network performance degradation and other situations. In the traditional network, sleep mechanism requires protocol and equipment support, it is complex and low economic benefit. Due to the use of hierarchical and centralized control, software-defined network(SDN) can easily achieve a variety of network applications. The paper realizes the smooth transition process of sleep mechanism of link based on SDN framework. The results show that the design ensures the performance and normal operation of the network while realizing dormancy.

1. Introduction

With continuous development of the network, more and more network equipment access to the network, resulting in a huge energy consumption[1], more and more attention has been paid to green network, and that how to reduce energy consumption is also an important consideration in network design. At present, the management of energy in the network is divided into two categories: sleep mechanism(SM) and link rate adaptive mechanism(LRAM)[2]. SM leaves the low load in sleep mode, so that energy consumption of them can be negligible, LRAM can dynamic adjust the rate of link according to network load. Compared to SM, in realization of LRAM, it needs to design complex protocol or update devices and it will bring huge economic costs[3], so in energy management, SM is widely used.

The traditional network has problems of closed structure, difficult expansion and difficult to deploy new application[4], although SM is easier to implement than LRAM in traditional network, SM is still limited to the structure. After 30 years of rapid development, many new network architectures are proposed, Software Defined Network(SDN) is one of those[5]. In traditional network, control plane and forwarding plane are centralized on one devices; SDN separates the control plane and the forwarding plane, all devices are controlled by the centralized controller[6-7]. As with the traditional network, SDN also concerns network energy consumption and SM is also used by SDN to achieve energy management.

SM may changes the topology of network and would affect data forwarding, that how to deal with SM has significant effect on result of SM. If design of SM is unreasonable, SM can cause a sudden decline in network performance and may eventually affect the normal operation of the network.

2. Preliminary

In 2009，professor Mckeown formally put forward the concept of SDN. Using the hierarchical idea, SDN separates the control plane and data plane of the switch network, moves the control functions to the controller and provides the application programming interface to the upper layer, the control plane can get the details of running status of the underlying network devices by OpenFlow[8] protocol. By abstracting the Layer 2 forwarding and Layer 3 routing table mechanism as flow tables, the control plane can control the forwarding layer by issuing rules to forwarding layer, at the same time, control plane provides interfaces to application plane to operate network devices and that gives the application plane good platform.

SM is a kind of energy management technology which is often used in green network. In traditional SM, component state is just divided into working and sleeping state, the energy consumption of working state is the rated energy consumption, and the energy consumption of sleeping state is 0. With continuous development of SM, as paper[9,10] describes, SM also can be divided in to different stages including deep sleep, light sleep and other stages. This paper choses the traditional SM to design smooth transition process of sleep state based on SDN framework.

3. Design

3.1. Analysis

SM is used to achieve energy management and green network, however, SM should not have adverse effects on network normal operation and performance, it needs a algorithm to realize the smooth transition process of sleep state based on SDN framework to put components into sleep.

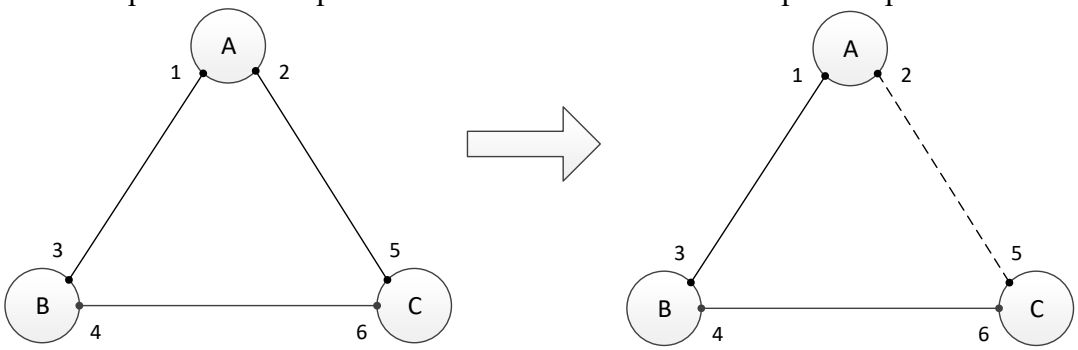

Fig 1. Demo of SM

As shown in Fig 1, the network is SDN network, A, B and C are switches those support OpenFlow protocol. As can be seen, link A-C is ready to be put into sleep state to save energy. Though link A-C has low load, link A-C may still have data stream; in port 2 and port 5, much data may still be in the queue waiting for forwarding, meanwhile, data is likely to be forwarding according to the original routing rules which are calculated by the left topology in Fig 1. If link A-C is put into sleep state in sudden, much data will lose and can not reach the destination node. So it is very important to design the process of sleep to prevent the occurrence of packet loss and network performance degradation.

Accompanied by the transition from the left to the right of the network topology in Fig 1, link A-C changes its state from working to sleep. To ensure the smooth transition process, the data packets that are forwarded according to the original routing rules in the topology have been forwarded completely and a mechanism is needed to be able to guarantee this process.

In SDN network, the underlying devices forward data according to flow tables issued by controller, the controller control underlying network by issuing routing rules in form of flow tables[11]. Structure of a flow table is shown in Fig 2, it consists of header, counters and actions. header field has many match fields, data can match the fields with IP, destination address, MAC address and etc[12]. After a data packet reaches a switch, the data will match flow table's match fields, if data matches one of the match fields, the data will perform actions following the actions filed of the flow table.

Fig 2.Structure of flow table

A switch has many flow tables, each flow table has its own priority and data packets match with flow table in order of priority from large to small[13]. The process is shown in Fig 3. As can be seen from Fig 3, the packet tries to match flow table in order of priority, if one of tables can be matched, the packet will execute actions(forward, drop and other actions).

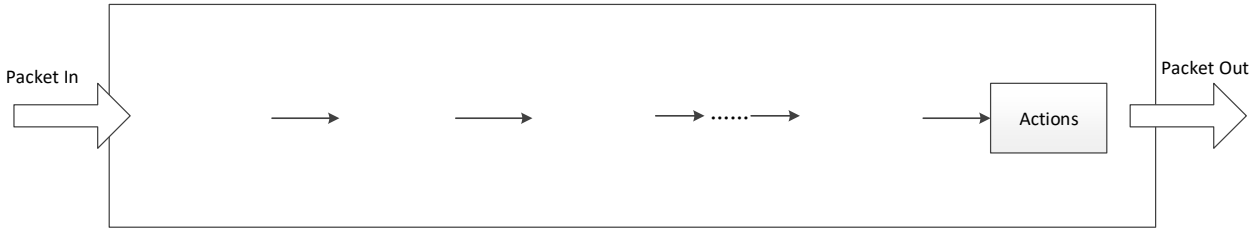

Fig 3. Packet pipelining process

3.2. Algorithm

To ensure the smooth transition process of sleep state, the key is that the sleep process need to wait some tome to ensure the data packets that are forwarded according to the original routing rules in the topology have been forwarded completely. As Fig 1 shows, transition from left to right topology must wait some time.

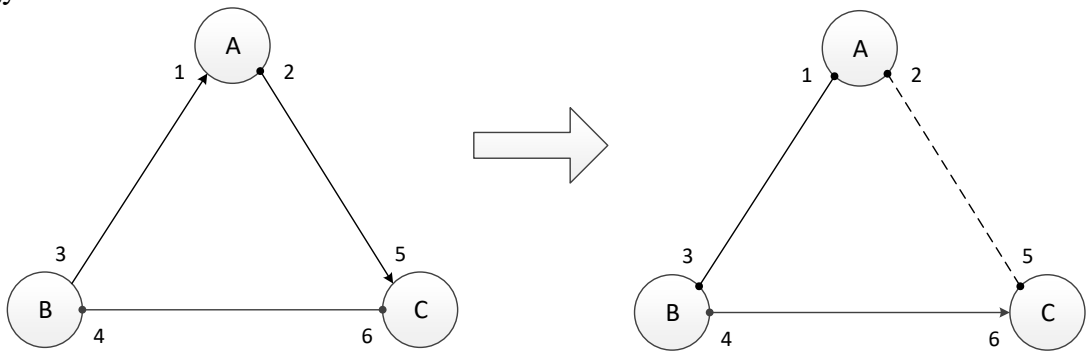

Fig 4. Demo of the change of route

As shown in Fig 4, in left topology, if a data packet needs to be transferred from B to C, it go through A, however, if link A-C is dormant, the packet has to be transferred for B to C directly. The

process will be analyzed in detail to explain how the left topology wait some time to leave link A-C sleep, at same time, it will give the algorithm that achieve smooth transition process of sleep state based on SDN framework.

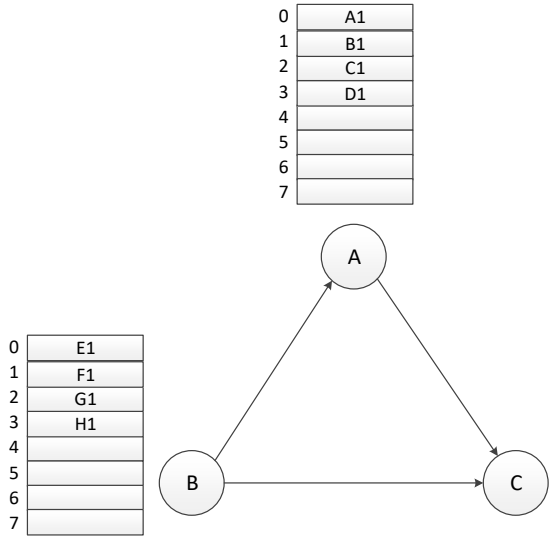

Fig 5. Beginning state of network

As shown in Fig 5, to meet the demand that data packets need to transfer from B to C and would go through A, A has four flow table, namely A1, B1, C1 and D1 and their priority order is A1, B1, C1, D1.

When the network need to put link A-C into sleep mode, link A-C is still at work and will wait some time.

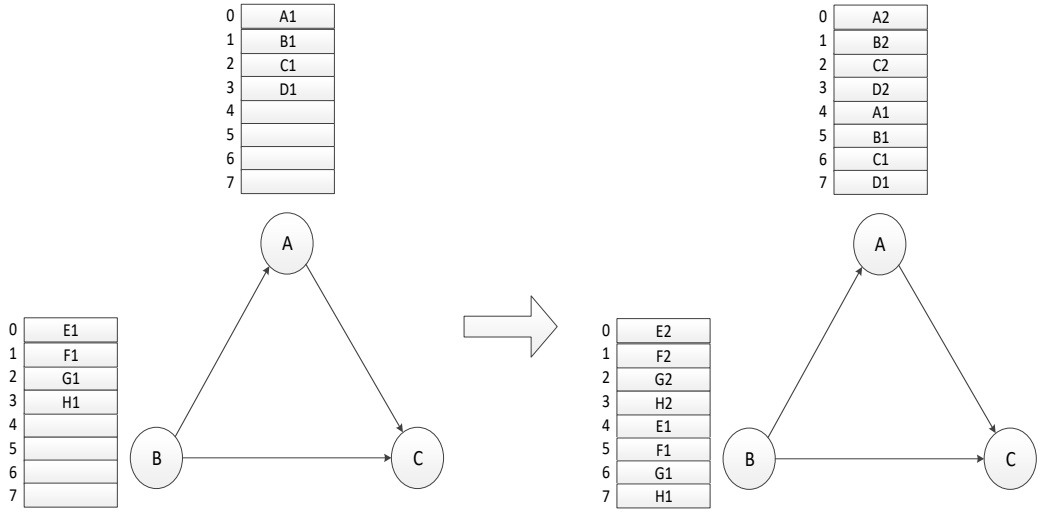

Fig 6. First stage of preparing for sleep

As shown in Fig 6, link A-C prepare to sleep, but it not immediately sleep. Assume the minimum priority of flow tables is P_{min}, the SDN controller set the flow tables A1, B1, C1, D1 the priority P_{min}-3, P_{min}-2, P_{min}-1, P_{min}, and it also need to record the number of flow tables it has operated N_f. New priority order will make the newly coming data packets match new flow tables while guaranteeing that old packet data can be forwarded. Because that the link A-C will sleep, data packets that transfer from B to C have to be forwarded to C directly, the SDN controller computes the new routing rules and issue new flow tables to A and B. With the new flow tables, new coming data packets from B to C will be transferred from B to C directory, meanwhile, the old data packets

transferred from B to C before link A-C need to sleep still use path B-A-C. So as can be seen from Fig 6, there are two paths that can forward data packets from B to C.

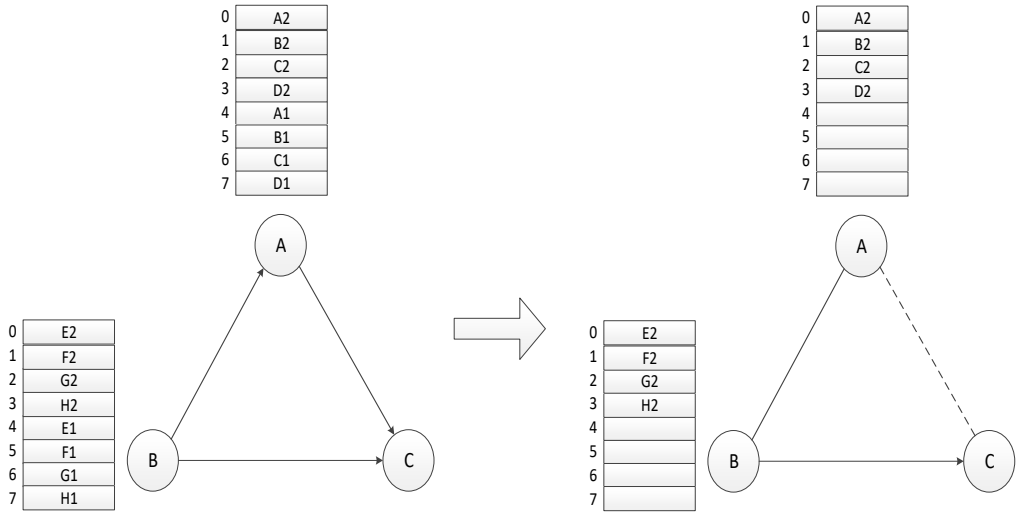

Fig 7. Second stage of preparing for sleep

The new flow tables and new priority order make it possible that all data packets can be forwarded. We know that link A-C will sleep finally and flow tables including A1, B1, C1, D1, E1, F1, G1 and H1 will be useless, it is necessary to know if there is any old data that match the old table. If no old data packet match the old table, the table can be deleted by controller.

As Fig 2 shows, any flow table has counters, the counters can record how many data packets the flow table has matched. So the SDN controller can scan switches whose flow tables has been degraded the priority, in each switch, the controller will scan N_f flow tables according to the priority from small to large. The controller may scan once a minute, if counters of a flow table did not change between the two times, the controller will delete the flow table and N_f of the switch needs to minus one. Until each N_f is zero, the SM process finishes, the resulted topology can be seen in Fig 7 and smooth transition process of sleep mechanism based on SDN framework has achieved.

4. Experiment

4.1. Software environment

The experiment was done in a simulated SDN environment and simulated SDN environment is composed of floodlight[14] and mininet[15]. Floodlight is a popular SDN controller, it can control the SDN network flexibly, mininet is a lightweight software defined network system platform, while providing support for the OpenFlow protocol, it can simulate the real environment of the SDN network.

Procedure of experiment:

(1)Setting up a topology in simulated SDN environment randomly as Fig 8 shows, bandwidth of each link is shown in Table 1.

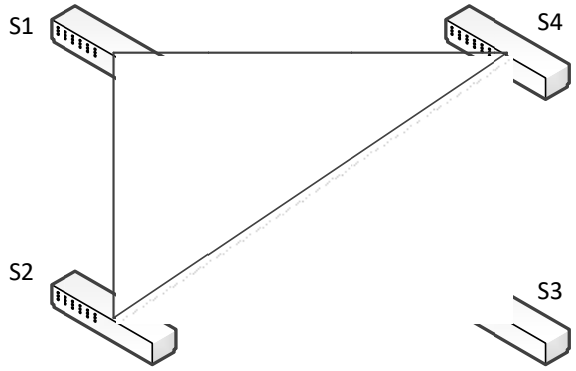

Fig 8.Topology of experiment

Table 1. Bandwidth of links

Link	Bandwidth(Mbps)
s1-s2	0.1
s1-s4	0.8
s2-s3	0.7
s2-s4	1
s3-s4	0.9

(2) Use Ping to test connectivity between s2 and s4, and record TTL.

(3)Let link s2-s4 be dormant directly, TTL between will be shown in left of Fig 9.

(4)Let link s2-s4 be dormant according to the algorithm, TTL between will be shown in right of Fig 9.

(5)Reset the network and use ping to test connectivity between s3 and s4, and record TTL.

(3)Let link s3-s4 be dormant directly, TTL between will be shown in left of Fig 10.

(4)Let link s3-s4 be dormant according to the algorithm, TTL between will be shown in right of Fig 10.

4.2. Results

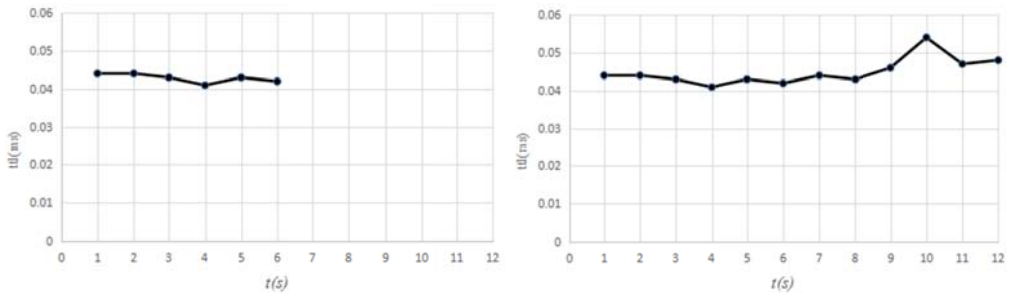

Fig 9. TTL between s2 and s4

As left of Fig 9 shows, if link s2-s4 sleep directly at sixth minute, packets can not be transferred from s2 to s4, TTL can not be computed, however, if link s2-s4 sleep according to algorithm in section 3.2, as left of Fig 9 shows, packets can still be transferred from s2 to s4 though TTL has some increase. The process of sleep is smooth and the network work normally.

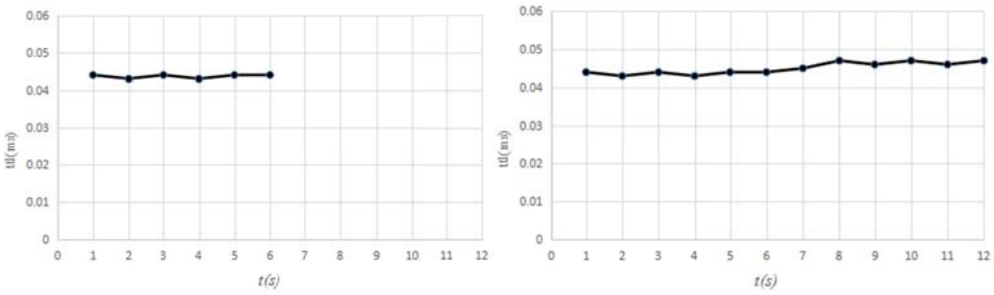

Fig 10.TTL between s3 and s4

As left of Fig 10 shows, if link s3-s4 sleep directly at sixth minute, packets can not be transferred from s3 to s4, TTL can not be computed, however, if link s3-s4 sleep according to algorithm in section 3.2, as left of Fig 10 shows, packets can still be transferred from s3 to s4 though TTL has some increase. The process of sleep is smooth and the network can work normally.

5. Conclusion

Sleep mechanism(SM) is a popular in network energy management, it can bring significant energy saving. Due to making components sleep, SM may cause packet lost and eventually lead to a decline in network performance, it is very important to make SM process smooth. In this paper, a mechanism for smooth transition process of sleep mechanism based on SDN framework is proposed, it has been proven that this mechanism is effective and practical. As the mechanism is based on SDN framework, it also brings new possibilities for the application of SDN.

6. Future work

As a SDN network may have a lot of switches, if controller scan all switches, it is time-consuming and would degrade the performance of centralized controller, so next work is to apply the mechanism in multi-controller SDN network and assign the job of scanning to each controller, meanwhile, it is also important to make controllers work together.

References

[1] Zhang F, Antonio F A, Hou C Y, et al. Network energy consumption models and energy efficient algorithms[J]. Chinese Journal Of Computers,2012,35(3): 603-615.

[2] Ling C, Tian Y, Yao M. Green Network and Green Evaluation: Mechanism, Modeling and Evaluation[J]. Chinese Journal Of Computers,2011,34(4):593-612.

[3] Aruna P B, Claude C, Dario R, et al. A Survey of Green Networking Research[C]. Communications Surveys Tutorials, IEEE, 2012, 14(1):3-20.

[4] Zhang C K, Cui Y, Tang H Y, et al. State-of-the-Art Survey on Software-Defined Networking(SDN)[J]. Journal of Software, 2015,26(1): 62-81.

[5] Mckeown N. Software-Defined networking. In: Proc. of the INFOCOM Key Note, 2009. http://infocom2009.ieee-infocom.org/technicalProgram.htm.

[6] Duang T, Lan J L, Cheng G Z, et al. Functional composition in software-defined network based on atomic capacity [J]. Journal on Communications,2015, 36(5):156-166

[7] Heng H, Huang W, Li T, et al. Multilevel DDoS protection mechanism based on SDS framework [J]. Computer Engineering and Applications, 2016, 52(1): 81-88.

[8] Erickson D. The Beacon OpenFlow controller. In Proc. HotSDN, Aug. 2013.

[9] Liu B, Lin C, Jian g X, et al. Performance analysis of sleep scheduling schemes in sensor networks using stochastic Petrinet. //Proceedings of the International Conference on Communications (ICC. 08) . Beijing, China,2008: 4278 -4283.

[10] Gupta M , Grover S, Sin gh S. A feasibility study for power management in LAN switches[C]. // Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP. 04) . Berlin, Germany, 2004: 361 - 371.
[11] Wang R, Bai B, Xu G, et al. Survey of energy-conservation technology in software-defined networking[J]. Application Research of Computers, 2016, 33(5):1285-1292.
[12] Sezer S, Scott-Hayward S, Chouhan P K, et al. Are we ready for SDN? Implementation challenges for software-defined networks[C]. Communications Magazine IEEE, 2013, 51(7):36-43.
[13] Shi S P, Zhuang L, Yang S J. SDN Optimization Algorithm Based on Prediction and Dynamic Load Factor[J]. Computer Science, 2017, 44(1):123-127.
[14] Floodlight OpenFlow Controller. http://floodlight.openflowhub.org/.
[15] Lantz B, Heller B, McKeown N. A network on a laptop: Rapid prototyping for software-defined networks. In Proc. of 9th ACM Wshop. on Hot Topics in Networks, 2010.