Fast Failure Recovery in Software-Defined Networks

Gang-Song Dong [1, a], Jing Shen [2,b] and Li-Qian Sun[3,c]

[1] Henan Electric Power Co., Ltd. Zhengzhou, China

[2] Henan Electric Power Co., Ltd. Zhengzhou, China

[3] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

[a] gang_songdong@163.com, [b] shenjingballball@163.com, [c] lqsun@bupt.edu.cn

Keywords: fast failure recovery; SDN; active; reactive.

Abstract. In Software defined networks (SDN), the control plane and the data plane are separated. One or more controllers can program a router or switch directly. In an in-band SDN network, the control flow (flow to or out of the controller) and the data stream share a transmission path. Thus, when the fault occurs on the path of the data stream, both the control flow and the data stream are affected. In order to achieve the operator's transmission quality requirements, the network should be able to recover within 50ms. This paper studies the fast fault recovery mechanism in SDN networks. For the control flow and data flow, we have studied the corresponding response and active recovery mechanism, and carried out the simulation experiment. The simulation results show that the reactive recovery mechanism can’t guarantee the failure to recover within 50ms, and the response mechanism of the control plane can cause the delay of data plane fault recovery. The simulation results also show that active recovery of control flow and data flow can meet the requirements of operator network recovery, even in large-scale networks with more traffic.

Introduction

Compared to OpenFlow switches, traditional switch designs are inefficient, complex, and expensive, it limits the innovation of network operators, researchers, and vendors. OpenFlow[1] solves this problem by isolating control plane and data plane. It removes control functions from the switch and puts them into one or more controllers. In this way, the forwarding function can be only being achieved in the hardware, if a new function wants to be added, it only needs to program on the controller to achieve. This design makes the network more flexible and easy to manage.

In order to achieve this requirement in the SDN, the fault in the control plane and the data plane must be restored within 50ms. Fast fault recovery is important for the control plane and the data plane. Because the data plane interrupt will also lead to control plane interrupt, and control plane interrupt will prevent the creation of any new flow of the switch affected by the failure.

Therefore, we study the reactive and active recovery mechanisms in SDN in order to meet the requirements of the operator's network. There are two kinds of SDN networks: in-band and out-of-band. In the in-band network, the control flow and the data flow share the communication channel (Figure 1A), while in the out-of-band network, the control flow and data flow use different channels (Figure 1B). The out-of-band network is expensive to establish because each switch requires additional physical ports, while in the in-band network does not require. The flow of incoming and outgoing controllers can be managed through the OpenFlow network (see Figure 1A).

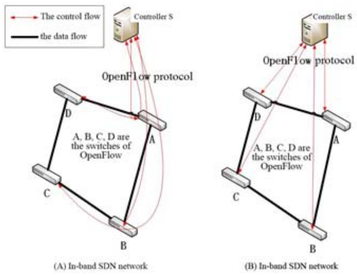

Figure 1 SDN network：
A.in-band network;
B. out-of-band network

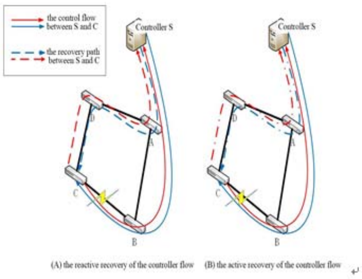
the control flow between S and C
the recovery path between S and C
Controller S
B
C
(A) the reactive recovery of the controller flow
(B) the active recovery of the controller flow

Figure 2 the fault recovery of the control plane

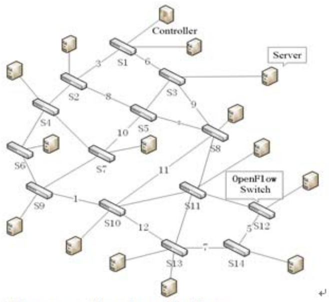

Figure. 3 simulation topology:backbone topology of SDN network

In this paper, we mainly study the fault recovery mechanism of the in-band network, and proposed reactive and active recovery technology, and its application in the control path and data path. We have completed an extended fault recovery experiment, and the results show that the recovery technology could not recover the fault within 50ms. Moreover, the reactive recovery mechanism of the control flow will delay the recovery of the data flow. The simulation results show that the active recovery mechanism on the data flow and the control flow can meet the operator's fault recovery requirements and can be applied to large networks with multiple flows.

SDN Fault Recovery

Network fault recovery refers to the network after the fault recovery to normal operation state. Firstly, this paper analyzes the fault recovery mechanism applied in the network and satisfies the requirements of the operator. And then these two mechanisms are integrated and applied to the SDN network to achieve millisecond recovery time.

Loss of Signal (LOS) and Bidirectional Forwarding Detection (BFD) are commonly used fault detection methods. LOS can detect faults on a specific port of a forwarding device, and BFD detects faults on the path between two forwarding devices. BFD uses a simple Hello protocol, and in many ways, is similar to many routing protocols of detection components, such as Open Shortest Path First (OSPF) protocol. The two systems (end-to-end devices) transmit BFD packets periodically, if a system no longer receives BFD packets, then the forwarding path between the two systems is considered to be faulty.

when a fault occurs, additional signaling is required to establish a recovery path. However, in the case of an active type, the recovery path is always calculated in advance and is retained. Therefore, when a fault occurs, no additional signaling is required to establish a protection path, and the packet will directly move to the recovery path.

In OpenFlow, faults can be detected by LOS. When the port state changes, the OpenFlow switch sends a port status message to the controller. This mechanism only detects local link failures and can be used in reactive recovery. However, for path protection, end-to-end fault detection is required in forwarding switches on any path. Therefore, we use BFD to detect faults in active recovery.

A Reactive Recovery Mechanism of Control Flow

The problem with control flow recovery is that the controller must establish a recovery path with the lost switch.

In the topology shown in Figure 1A, the control paths for switches B and C will be affected if the fault occurs on link AB (instead of link BC). We assume that the controller has established a working path similar to switch C for switch B, as shown in Figure 2A. Since the recovery of switch B needs to establish the recovery path (<SADCB> and <BCDAS>) through switch C, the controller must first store the control flow of switch C before attempting to recover switch B. However, the controller does not know when the recovery work of switch C is completed. To understand this, we use the concept of barrier requests and recovery messages, which are described in the OpenFlow white paper[2]. Therefore, the controller sends all the flow-mod messages to the switch A and D (about the recovery operation of the switch C) and then sends a barrier request message. After this, the controller waits for the execution of the recovery process of switch B until it receives the failback message from switches A and D. These failback messages confirm that the recovery process for switch C has been completed.

B Active Recovery Mechanism of Control Flow

Data may be lost during fault detection and recovery. In order to further reduce the data loss caused by the recovery process, we can adopt active recovery mechanism. Active recovery does not require the OpenFlow switch to contact the controller to modify or add a flow table entry to establish an available path.

We propose the active recovery mechanism for control flow in the in-band SDN network using the concept of group table proposed in OpenFlow1.1..

The active recovery mechanism for 1:1 of the control flow of switch C, as shown in Figure 2B. The execution operation associated with the next execution block of the outgoing port D (for switches A and C) is executed. Because the flow table entries of the path<SADC> in switches D and C and the associated path of switch D and A already exist, there is no need to add these flow table entries when a fault occurs.

C Reactive and Active Recovery Mechanisms for Data Flow

In the reactive recovery of a data flow, an available path is established by adding and modifying the flow table entries in the switch on the available path. In order to establish the available path of the data flow, the fault recovery of the control flow is critical.

For active recovery of data flow, the available paths are established before the failure occurs. When the working path fails, the ingress OpenFlow switch will redirects the flow to the available path through the group table. Because the controller is not involved in the recovery of data flow, the fault recovery of control flow is not affected by the recovery of data flow.

Simulation Experiment

In this section, we introduce the experimental environment, simulation tools, topology and experimental methods.

A Simulation Tools and Topology

We do experiments on a virtual machine with Ubuntu12.04. In order to simulate the OpenFlow Virtual Switch [3] and the backbone network topology.

We use Mininet [4] to generate the backbone network topology required for the experiment (see Fig. 3). In Fig. 3, each switch in the topology is connected to the server, and the S1 switch is directly connected to the controller. In order to assign an independent CPU cores to the controller and switch, we have different cores for the nodes that are connected to the controller and each switch.

B Simulation Method

In the simulation, each server generates packets to other servers, as shown in Figure 3. In order to transfer packets, we manually configure the routing tables and ARP tables on each server. According to the reactive and the active recovery of control flow and data flow, we did the following four experiments: reactive, active-reactive, reactive-active and active.

In each of the above tests, a link between the switches will fail and the recovery time will be counted. In the simulation, each switch in the backbone network topology will first establish an OpenFlow session with the controller through the algorithm implemented in [5].

C Simulation Results Analysis

In this experiment, the link S1-S3 in the topology is faulted firstly. Figure 5 shows the flow of links S1-S3. This link is the only link to S1 after the link S1-S3 failed. This can be seen from 0s to 0.05s in Figs. 5A and 5B and from 0s to 0.04s in Fig. 5C and Fig. 5D. After the decline, the total flow in the two link of S2-S1 and suddenly increased. In Figure 5A and Figure 5B, it is because of the reactive recovery operation of the controller. In Figure 5C, it is because of the reactive and active recovery operations. In Figure 5D, it is because of the active recovery operation.

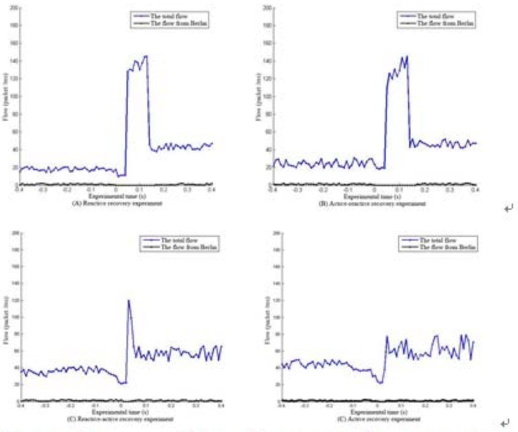

Figure 5 The flow on the reactive / active link

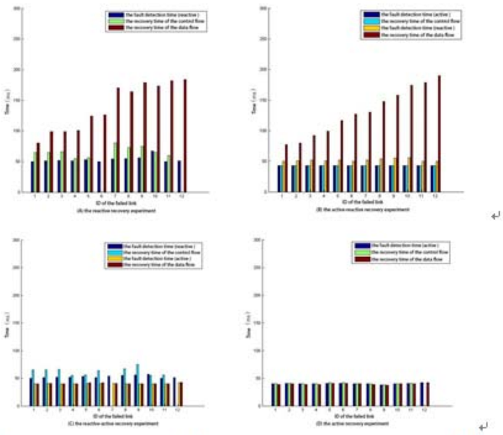

Figure 6 the recovery time of data flow and control flow

Figure 5 shows that there is a gap in the flow from the S3 when the link S1-S3 fails, indicating that the data flow is interrupted when the fault occurs.

Figure 6A shows that the recovery time of the data flow does not change linearly as the number of affected flows increases. In Figure 6B, the recovery time varies linearly as the number of affected flows increases. However, in Figures 6C and 6D, the recovery time of the data flow remains unchanged as the number of affected flows changes.

Conclusion

In this paper, we propose a reactive and active recovery mechanism for control flow and data flow, and perform fault recovery experiments in in-band SDN networks. Experiments show that OpenFlow can recover the fault, but cannot meet the recovery time requirement of 50ms in large-scale network. Moreover, the recovery of the control flow causes the recovery of the data flow to be delayed. The simulation experiment also shows that the active recovery of the control flow and data flow can meet the 50ms recovery time requirement, even in the large-scale network.

Acknowledgements

This work was financially supported by Henan electric power company science and technology projects: research and application of Henan power communication backbone network modeling based on simulation technology (SGHAXT00GCJS1600133).

References

[1] N. McKeown, T. Andershnan, H. Balakrishnan, G. Parulkar, L. Peterson,J. Rexford, S. Shenker, and J. Turner, Openflow: Enabling innovation incampus networks, ACM Computer Communication Review, 2008.

[2] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester,Enabling Fast Failure Recovery in OpenFlow Networks, DRCN, 2011.

[3] J. P. Vasseur, M. Pickavet, and P. Demeester, Network recovery: pro-tection and restoration of optical, SONET-SDH, IP and MPLS, MorganKaufmann, 2004.

[4] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, Open-Flow: Meeting carrier-grade recovery requirements, Computer Communications, 2012.

[5] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester,A demonstration of automatic bootstrapping of resilient OpenFlow networks, IFIP/IEEE Integrated Network Management Symposium, 2013.