AN APPROACH TO PREVENT CASCADING FAILURES OF MULTI-CONTROLLERS IN SOFTWARE DEFINED NETWORKS

[1]Ashish Kumar Dhanotiya, [2]Saumya Hegde

[1] Scholar, [2]Assistant Professor, Department of Computer Science and Engineering,
National Institute of Technology, Surathkal, Karnataka (India)

ABSTRACT

Currently SDN designs for data centers are using multiple controllers to control the data center network and datacenter traffic. Since the traffic in the data center network is very huge, so a single controller cannot control the whole traffic, so to solve this problem some recent proposals suggested use multiple controllers. In this paper, we are focusing on the cascading failure of controllers. When a controller gets overloaded then it fails and switches under this controller will be assigned to another controller randomly, so the load of the failed controller is nonmoving to other controllers which may exceed the capacity of other controller and cause them to fail and then load of this controller also moves to other controllers and in a similar wait may cause failure of other controllers also, this will cause the cascading failure of the controllers. Here we are proposing away to prevent the controllers from cascading failures. Initially we will show a scenario for cascading failures of controller's and then propose a strategy to prevent cascading failure.

Keywords: Controller, Control Plane, Cascading Failure, Data Plane, Flow Rules, Reliability, Software Defined Networks.

I. INTRODUCTION

Present computer networks are huge and complex to control and manage, there are many equipment involved in computer networks like routers and switches, firewalls, network address translators, intrusion detection systems [1]. When a large number of end systems are added it becomes difficult to adjust the network infrastructure. SDN separate control plane from network devices. A centralized controller performs the control operations. Every time when a packet from a new flow comes to a switch, it contacts to its controller for flow rules. The Controller decides the rules to handle the packets and it gives instructions to the switches. And packets are forwarded by the switches based on the controller instructions.

Software Defined Networking gives hope to change the current network infrastructure limitations [1]. It decouples the control plane and data plane and converts the network switches to simple forwarding elements and a logically centralized controller implements the control logic. As it is clear that the whole network depends on the controller so if a controller fails SDN networks will not be able to forward the packets, so the reliability of the network is important, and it depends on the reliability of the controller. Since networks with one controller suffer from a single point of failure so to prevent this multiple controllers are used [2]. In multiple controller approach if one controller fails then other controllers are ready to take the responsibility of switches which were under the control of that failed controller.

So the reliability of SDN network is increased by the use of multiple controllers. These multiple controllers cannot assure the reliability of the network because if one controller fails, then its load is shared by other controllers, and there is no optimal strategy in between these controllers by which we can assign the switches of failed controller to these controllers so that it will not lead any further controller failure.

Load on the controller: We are considering load as a main parameter for the failure of the controller. In SDN, for load on controller we mainly consider the number of PACKET_IN messages or the number of flow requests and installing the flow rules. Heavily loaded controllers always have higher probability of failure as they have fewer resources to handle the load [4]. Sometimes the failure of a controller may cause cascading failures of other controllers [3].

Cascading Failure: If there is a system in which there are many parts and each part is interconnected and dependent on the other parts for their reliability and survival or we can say that every part is sharing some load of the system and ready to take the extra load if any part of the system fails. If a part fails and its load is taken by another part and causes failure of that part and this process continues to the failure of other parts and thus causes the failure of the whole system, this failure is known as cascading failure. Computer networks can also suffer from cascading failures. In computer networks the traffic is forwarded by routers and switches along appropriate paths. So if a router or a node overloaded then it causes the failure of that node or router or it can also be caused when a router or a node is taken down for maintenance so in both the cases the traffic is forwarded through another node which causes that node to be overloaded and thus that node fails, in this way the cascading failure of network occurs.

II. LITERATURE SURVEY

IP networks which we are using from long time are complex and hard to manage. To achieve the required high level network policies, network operators requires configuring and managing every individual network device separately using vendor-specific commands [1]. Currently the control plane (which decides the forwarding rules) and data plane (which forwards traffic according to the forwarding rules provided by the controller). It reduces the innovation and flexibility of networking devices [1]. Software Defined networking is a new way to overcome these limitations and it help to increase the speed of innovation. It separates the control plane and data plane so that both can evolve separately. By separating both the planes, switches acts only as forwarding devices, and forwards traffic according to the instructions of controller. The entire load is carried by the controller only. SDN is now widely used in current networks like WAN and Data center networks. In recent years data centers are increasingly deployed at various places. Since data centers have high traffic [7], so it is very difficult to manage that much traffic, it becomes very complex structure, and a single controller suffers from single point of failure problem, if the controller fails then whole network collapse. For managing this complex structure and preventing the network from single point of failure, multiple controllers are used [5]. These multiple controllers are distributed in the data centers. Every controller has some portion of the switches. So when a controller fails then also network works properly, only the switches under the failed controller get affected. These switches are then assigned to the remaining controllers. The main problem here is to place these controllers that are how many controllers are required and where these controllers should be placed [6]? If one controller fails then its load is handled by other controllers which may cause the failure of other controllers also [3]. In [3], author has proposed the problem of cascading failures of controllers in Software Defined Networking; here we are providing a solution to this problem.

III. MODEL FOR CASCADING FAILURE IN SDN:

In SDN controller is responsible for the whole network operations and switches only acts as forwarding devices which forwards data according to the instructions given by the controller, Currently for data center networks the traffic is huge, so multiple controllers are used to handle that traffic, every switch is assigned to exactly one controller. Whenever a new flow request comes to a switch, it asks to its controller to provide flow rule for that request, controller installs the flow rule along the path and then the traffic is forwarded, the time required to install the flow rule is known as flow setup time [2]. Thus the failure of a controller can cause the failure of the whole network assigned to it. In a current multi controller environment if one controller fails, then the switches under that controller are assigned to other controllers randomly, which may cause the failure of other controllerstoo, and this may lead the cascading failure of controllers, sothe whole network gets failed in this way[3].

In figure 1 cascading failure of controller is shown. [1] is showing when the network is working fine, in [2] the load on blue controller increases and it exceeds its capacity so the blue controller fails and its switches assigned to other controllers, [3] Now since switches of blue controller are assigned randomly so the load on green controller increases and exceeds to its capacity so green controller now fails, [4] Now the switches of the failed controllers are assigned to remaining red and yellow controllers and the load on red controller increases and exceeds its capacity so the red controller fails, [5] Now in similar way the last controller takes over all the switches and its load exceeds its capacity so it also fails and thus the whole network fails. As shown in figure 1, on the failure of one controller the load of that controller is distributed among other controllers. Currently we are having random assignment of switches so it is possible that all switches or most of the switches of the failed controller may be assigned to only a single controller and the load of the controller which takes the load of failed controller will increase. And it may increase in such a way that it can exceed to the capacity of the controller and causes the failure of that controller, in this way other controller may also fail, it is also possible that all the controllers fails and the whole network may collapse. It is easy to say that the probability of failure of an SDN network is high if the initial failed controller has the maximum load [3]

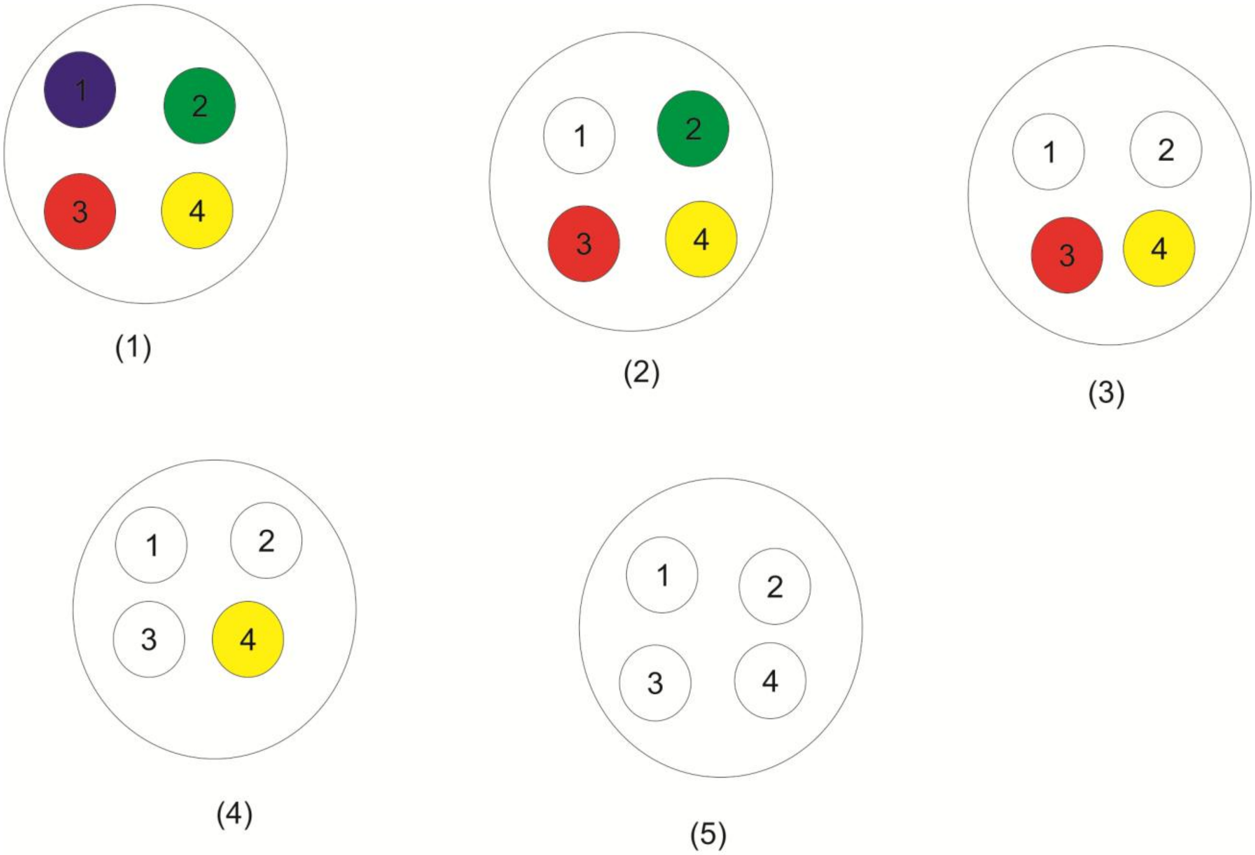

Fig.1 Cascading failure in Software Defined Networking

IV. PROPOSED SOLUTION

Currently we are not having any control over the assignment of new switches arriving at the SDN network and for the assignment of the switches of the initially failed controller. So here we are proposing a centralized controller by which we can control the assignment of switches. This centralized controller will only take care of the assignment of switches to the controllers in this way it will not have any traffic load.

In the figure2 it is shown that a single, centralized controller C is having control over all the controllers, this controller is aware of the load on each controller and their capacities. Here we are having two types of controllers' active controllers and inactive controllers. Those controllers which are having at least one switch assigned to them are known as active controllers and other controllers are known as inactive controllers. We are assuming that there are sufficient numbers of controllers to handle the whole load of the SDN network.

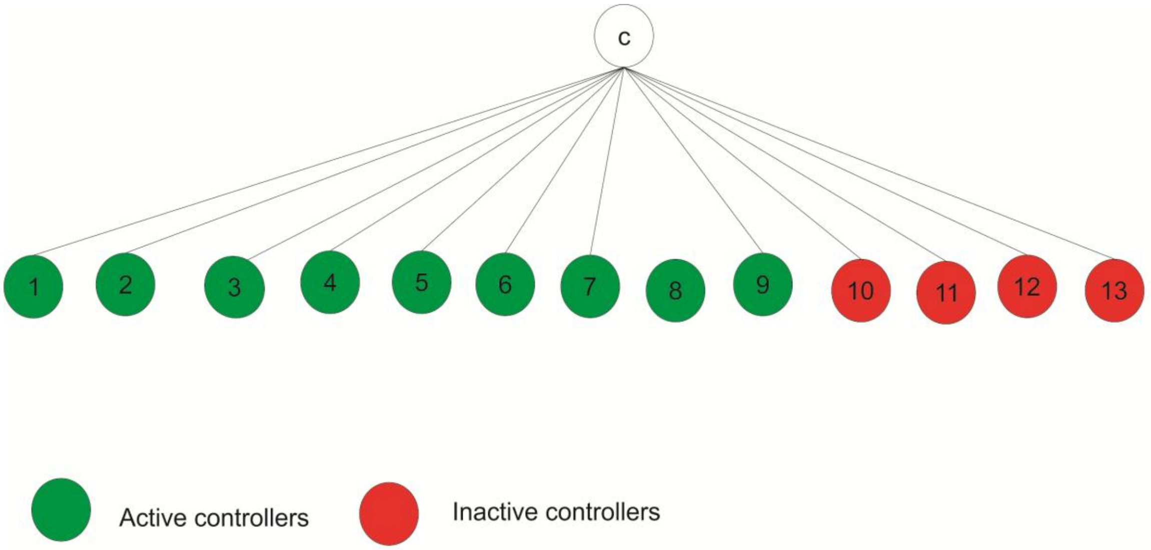

Fig.2 MultiController Environment with Single Centralized Controller

Here every time a new switch comes it requests to this central controller C for its assignment to a controller, now controller C will assign this new switch to a controller with minimum load.

And before assignment, it checks the load of the controller; if it is more than 80 percent of the capacity of the controllerthen it will not assign that switch to that controller because itmay exceed the capacity of the controller. So when centralized controllers do not find any active controller capable of handlingthe load of the failed controller it simply makes one of the inactive controllers to active and then assign switches to that controller.

When a controller in this SDN network fails then every switch under that controller will be considered as a new switch and then the centralized switch starts assignment of these switches to the controllers with minimum load and if load on each active controller exceeds more than 80 percent of its load capacity then centralized controller will make an inactive controller as an active controller and then assigns these switches to that controller, and there are not sufficient controllers then centralized controller will simply discards the requests of the switches and will not assign them to any of the controller to prevent the reliability of the controller.

V. EVALUTION

The Proposed strategy ensures the prevention of the cascading failure of the controllers as in this strategy no switch assignment is done if the load on the controller reaches near to its capacity. It is preventing the failure of initial controller, the only thing can happen is if load on a controller dynamically increases at any point of time and exceeds its capacity, then that controller can fail but it will not cause the failure of other controller. In proposed strategy, we are able to distribute load nearly in equal portions to the controllers so it will equalize and reduce the flow setup time.

This solution ensures the prevention of cascading failure of controllers by fulfilling following conditions:

1. It is ensuring that there are sufficient controllers with sufficient capacity to handle the load of the failed controller.
2. It is ensuring the balanced distribution of load among all the controllers to prevent the failure of the maximum load controller, as every controller has nearly same load.
3. The load distribution after the failure of any controller will not cause the failure of any other controller.

VI. CONCLUSION

In this paper we focused on the cascading failure problem of the controllers of the multi controller environment of data center networks when SDN is used in data center networks. Cascading failure of controllers may cause the failure of the whole data center network. Proposed strategy is well suited for preventing the cascading failure of the SDN network in data centers.

REFERENCES

Journal Papers:

[1] Diego Kreutz, Fernando M.V. Ramos, Paulo Verissimo, Christien EsteveRothberg, Siamak Azodolmolky, Steeve Uhlig *"Software-DefinedNetworking: A Comprehensive Survey," in Proceedings of the IEEE(Volume:103 , Issue: 1),2015, pp.14-76.*

[2] Md. Faizul Bari, Arup Raton Roy, Shihabur Rahman Chowdhury, QiZhang, Mohamed Faten Zhani, Reaz Ahmed, and Raouf Boutaba *"DynamicController Provisioning in Software Defined Networks," 9th CNSM and Workshops 2013 IFIP,*

[3] G. Yao, J.Bi, and L.Guo *"On the cascading failures of multi-controllersin software defined networks," in proc. IEEE ICNP, 2013, pp.1-2,*

[4] G. Yao, J.Bi, Y. Li, and L.*Guo "On the Capacitated Controller PlacementProblem in Software Defined Networks," IEEE Communications Letters,IEEE (Volume: 18, Issue: 8), pp. 1339-1342*

[5] *Kevin Phemius, Mathieu Bouet and Jeremie Leguay "DISCO: Distributedmulti-domain SDN controllers," Network Operations and ManagementSymposium (NOMS), 2014 IEEE, pp. 1-4*

[6] Brandon Heller, Rob Sherwood, Nick Mckeown *"The ControllerPlacement Problem," HotSDN'12, August 13, 2012, Helsinki, Finland.*

[7] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang*"Understanding Data Center Traffic Characteristics," ACM, WREN09,August 21, 2009, Barcelona, Spain.*