Journal of Global Research in Computer Sciences
2021
Vol.12 No.1

# A Hybrid Hierarchical Control Plane for Software-Defined Network

**Arpitha T[1], Usha K Patil[2]**
[1*]MTech Student, Computer Science & Engineering, GSSSIETW, Mysuru, India
[2*]Assistant Professor, Dept of CSE, GSSSIETW, Mysuru, India
[*]Affiliated to VTU, Karnataka

The decoupled architecture and the fine- grained flow-control feature limit the scalability of a flow-based software-defined network (SDN). In order to address this problem, some studies construct a flat control plane architecture, others build a hierarchical control plane architecture to improve the scalability of an SDN[1]. However, the two kinds of structure still have unresolved issues: A flat control plane structure cannot solve the super linear computational complexity growth of the control plane when the SDN scales to a large size, and the centralized abstracted hierarchical control plane structure brings a path stretch problem[2].

To address these two issues, an Orion[3], a hybrid hierarchical control plane for large-scale networks. Orion can effectively reduce the computational complexity of an SDN control plane by several orders of magnitude[4]. We also design an abstracted hierarchical routing method to solve the path stretch problem.

Furthermore, a hierarchical fast reroute method to illustrate how to achieve fast rerouting in the proposed hybrid hierarchical control plane. Orion is implemented to verify the feasibility of the hybrid hierarchical approach. The objective of this work is to combine the advantages of flat and centralized hierarchical control planes, and to address super linear computational complexity growth , path stretch problem of the control plane and to achieve fast rerouting hybrid hierarchical control plane.

Since the flat control plane architecture cannot solve the super-linear computational complexity growth of the control plane when a SDN network scales to large size.

## RELATED WORK

Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications For the realization of scalable Software-Defined Network having control over frequently occurring events on control plane is essential. One way to do this is to process those events in data plane itself which requires customization of switches which affects control plane in terms of visibility. In this paper author proposes Kandoo. It is a framework for achieving scalability without making changes to switches. Kandoo has two layers of controllers: (i) the bottom layer is a group of controllers with no interconnection, and no knowledge of the network-wide state, and (ii ) the top layer is a logically centralized controller that maintains the network-wide state. Design of Kandoo helps network operators to reflect local controllers on demand and ease the load on top layer. Maestro: A System for Scalable OpenFlow Control Controller is responsible for the initial establishment of every flow by contacting related switches is the primary characteristic of an OpenFlow network. Therefore performance of the controller could be a bottleneck. This paper shows how this fundamental problem is addressed by parallelism.

The state of the art OpenFlow controller, called NOX, achieves a simple programming model for control function development by having a single-threaded event-loop still NOX has not considered exploiting parallelism. Authors of this paper propose Maestro which is a programming model for programmers and also accomplishes parallelism in with additional throughput optimization techniques.

The Beacon OpenFlow Controller

Beacon is a Java-based open source OpenFlow controller created in 2010. It has been widely used for teaching, research, and as the basis of Floodlight. This paper shows the architectural decisions and implementation that realizes three of Beacon's goals: to improve developer productivity, to provide the runtime ability to start and stop existing and new applications, and to be high performance.

Scalable Flow-Based Networking with DIFANE

Administrators of enterprises can specify policies that drive how the underlying switches forward, drop, and measure traffic. . However, existing techniques for flowbased networking rely too heavily on centralized controller software that installs rules reactively, based on the first packet of each flow. Author of this paper proposes DIFANE, which is a scalable and prominent solution that keeps all traffic in the data plane by selectively routing packets through intermediate switches where necessary rules are stored. DIFANE assigns the controller to do the simpler task of partitioning these rules over the switches. DIFANE can be readily implemented with commodity switch hardware, since all data-plane functions can be expressed in terms of wildcard rules that perform simple actions on matching packets.

SDN decouples the network's control plane and data plane, and extracts complex control functions from network devices. It also supports a fine-grained flow-based management based on the OpenFlow protocol to enable highly programmable and flexible networks[5].

Currently, almost all commercial switches support the OpenFlow protocol and other south- bound interface protocols, such as Cisco onePK API, also support the fine-grained flow control feature of SDN[6]. Flow-based SDN, introduces a great communication overhead between the data plane and the control plane, which limits the scalability.

The Limitations of the existing system are :

1) Great communication overhead between the data plane and the control plane, which limits the scalability.

2) Computational complexity growth of the control plane when a SDN network scales to large size.

3) Centralized Hierarchical control plane architecture brings path stretch problem.

PROPOSED SYSTEM

The proposed architecture combines the advantages of flat and centralized hierarchical control planes, and addresses the two unresolved problems specified above.

First we design Orion, a hybrid hierarchical control plane which can reduce the computational complexity growth of the SDN control plane by constructing abstracted hierarchical network views[10].

Second, the design is an abstracted hierarchical routing method to address the path stretch problem by constructing abstract intra-area links and pre-calculating all intra-area abstracted links hops. Third, proposes a hierarchical fast reroute method to illustrate how to achieve fast rerouting in the proposed hybrid hierarchical control plane.

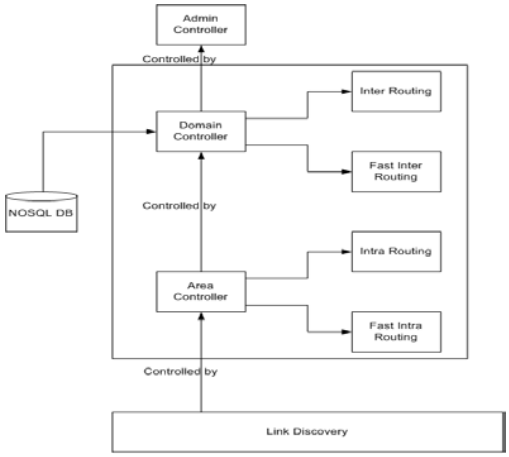


Fig.1. Hierarchically structured control planes in SDN environment

The control planes are structured as shown in the figure 1. Fast intra routing is managed by the area controller whereas inter routing is handled by domain controller. Finally all the management issues will be taken care by admin controller.

Link Discovery Module: This Module obtains intra-area and inter-area link information through the Link Layer Discovery (LLDP) protocol. When there are multiple areas, the

module needs to acquire LLDP messages from other areas to discover the links. To obtain inter area link information, one area controller sends LLDP packets to all ports of its edge switches. Upon receipt of a packet the switch forwards it to the edge switch in another area through their physical link.

Area Controller: This Module controls the routing between intra routing and fast intra routing.

Domain Controller: This Module controls the routing between inter routing and fast inter routing. This module controls the Area controller.

Admin Controller: This module controls the Domain controller.

The links and switches generated were stored in a database called nosql db here.

Mininet is used in the Link Discovery module. OpenDayLight is used during domain controller module. Advantage of proposed system is that Orion can effectively reduce the computational complexity of a flowbased SDN control plane by several orders of magnitude, and solve the path stretch problem brought by the logical hierarchical control plane architecture.

RESULTS and DISCUSSIONS:

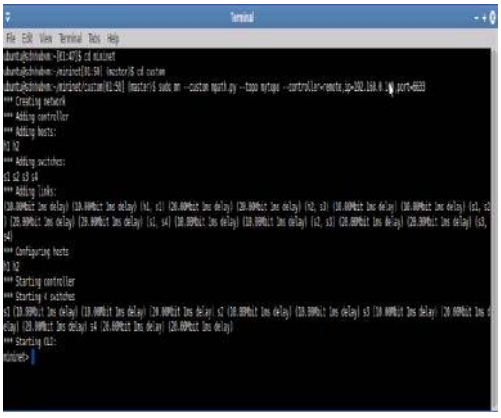

Figure 2: Adding hosts and switches to define area and establishing links in mininet

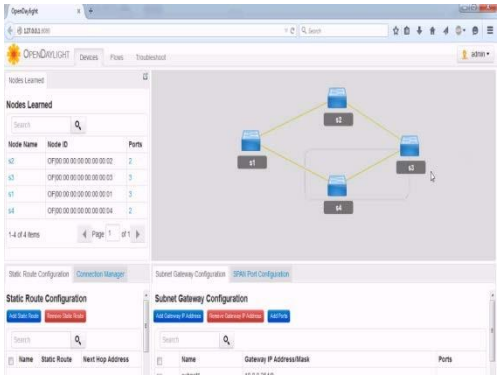

Figure 3: Realization of switches and links in SDN controller

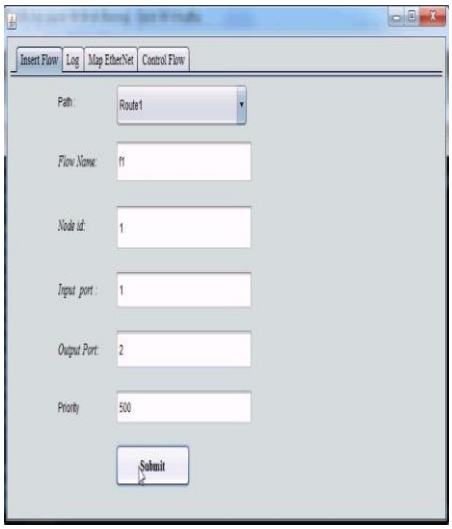


Figure 4: Assigning particular node id, input port and output port for the routing

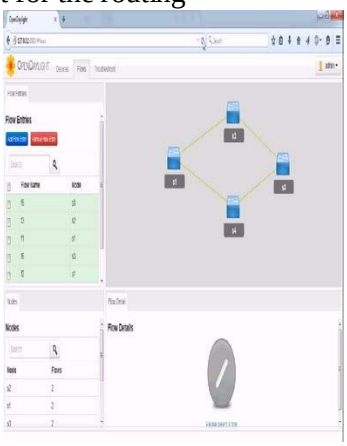

Figure 5: Routing with particular source and destination address

Starting from configuring mininet which is an emulator and OpenDayLight which is the SDN controller, switches and links are added in the mininet. The areas can be defines according to the number of hosts present within each area. Then in the next step those switches and are added in SDN controller also. The routing is made through ping between to hosts. If routing is within the area then it is referred as intra routing whereas if routing is between two areas the it is referred as inter-routing.

Failure of links within particular area is handled by

The area controller and inter-routing failure is handled by domain controller and finally everything is managed by Admin controller that is SDN controller. If the communication between two hosts is marked as failure, then controller looks for the alternate path with the minimum bandwidth and communication is established. In this way link failure is handled by area controller as well as domain controller.

## CONCLUSION

Hybrid hierarchical control planes in Software defined network addresses the scalability issues for larger network as well as routing and re routing issues. Two concepts are put together in this work that is SDN with multiple control planes and those controllers are arranged in hierarchical fashion. Having multiple control planes eases the burden of single control plane and addresses scalability as well.

## REFERENCES

[1] C. Fraleigh and et al, "Packet-level traffic measurements from the sprint ip backbone," Network, IEEE, vol. 17, no. 6, pp. 6–16, 2003.

[2] D. Murry and et al, "The state of enterprise network traffic in 2012," in Communications (APCC), 2012 18th Asia-Pacific Conference on. IEEE, 2012, pp. 179–184.

[3] M. Zhang and et al , "Analysis of udp traffic usage on internet backbone links," in Applications and the Internet, 2009. SAINT'09. Ninth Annual International Symposium on. IEEE, 2009, pp. 280–281.

[4] W. John and et al, "Analysis of internet backbone traffic and header anomalies observed," in Proceedings of

the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007, pp. 111–116.
[5] M. Yu and et al , "Software defined traffic measurement with opensketch," in Proceedings 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI, vol. 13, 2013.and Service Science (ICSESS), 2015 6th IEEE International Conference on. IEEE, 2015.