International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 08, August 2016
www.ijates.com
ijates
ISSN 2348 - 7550

CONGESTION AWARE ROUTING IN SOFTWARE DEFINED NETWORK (CARS)

Santosh Kumar[1], Ramphal Maurya[2]

[1,2]Student M.Tech CSE Department, MNNIT Allahabad University (India)

ABSTRACT

Todays computing world mostly relies on Internet for its day-to-day activity. Everything is connected and accessible from anywhere with the help of Internet. Despite widespread adoption, Tradition IP network is very hard to manage. It is very difficult to configure existing network and reconfigure to respond the faults, load and changes. Managing the Internet's core has become very complex and tedious and error prone. The main reason behind it is tremendous growth in Internet users and large network traffic they generate. In this paper, proposed methodology describes the process of automatically find alternate path in network by identifying network traffic and congestion in the current path and finally switching off the appropriate nodes or links. With a specific end goal to compute the QoS routes, it is key to gather up-to-date network state information, for example, delay, transfer speed, and packet loss rate for every connection.

Keywords: SDN, Open Flow, QoS, Floodlight controller, Traffic Engineering.

I. INTRODUCTION

As computer networks continue to expand across the globe, data needs to travel a farther distance between endpoints. Internet traffic volume continuously growing network transfer information between users, increase in demand of resources and height quality of services according to application needs makes network congested, resulting in higher round trip time (RTT) and packet loss. For the network it is important to avoid traffic. Software Defined Networking emerges as new era in computer networking leveraged as a solution for large complex network and seamless throughput over network [6]. Network traffic management technique exists but that is very tough work in current networking scenario. Many router are distributed and located at different place all have their unique structure. It is very tough work for network administration to change the network according to our traffic need. These entire problems can be solved by controlling whole network from one place, but in future if resources and traffic demand increases one controlling system will not work that will get congested.

Software Defined Network SDN is described by partitioning architecture into three layers [1]. All layers have unique functionality and relationship with upper and lower layer. These layers are control, data and application plane. SDN is new era in internet where network is programmable. A network administrator can change routing and rule for whole network dynamically. Current networking is mostly depends on hardware devices called vender dependent that leads to lot of problems if we want to change our routing algorithm or want to expand our network. SDN have all the solution of problem arises in current network it extract all controlling power of network switches to a one place called central server and now switches act as dumped device and only work is to transfer the incoming packet from port to output port. Fallowing the three major part of SDN is discussed below.

1. **Control Plane:** SDN control plane is centralized control software running at server. Control plane enable application administrator to apply new policies and protocol to data plane and make decision about where traffic should flow. NOX, POX, Floodlight and OpenDaylight etc. are examples of SDN controller running on control plane.
2. **Data Plane:** Data plane is comprises of forwarding devices carries network traffic. Forward packet to next hop according to control plane rule.
3. **Application Plane:** SDN Applications are programs that explicitly, specifically, and programmatically communicate their network necessities and desired network behavior to the SDN Controller via NBIs [1].

Open Flow is SDN protocol that is used to centrally control network forwarding devices. This is first standard protocol for the communication between control plane and data plane. Open Flow [2] gives programming based access to the flow tables present in forwarding devices that educate routers and switches how to direct network traffic movement. Open Flow protocol handles by Open Networking Foundation. Protocol provides set of management tools to handle topology changes. Open Flow hybrid switches support both Open Flow operations and common Ethernet switching operations [3]. An Open Flow Switch communicates with SDN controller by Open Flow protocol Consists of a group table and one or more Open Flow tables, which perform packet lookups and forwarding, and one or more OpenFlow channels to the outer controller. Figure shows the general structure of communication made by SDN controller to all data devices present at data plane. For reliability and security Open Flow run over TCP and uses SSL security. Figure 1 shows simple Open Flow Protocol function.

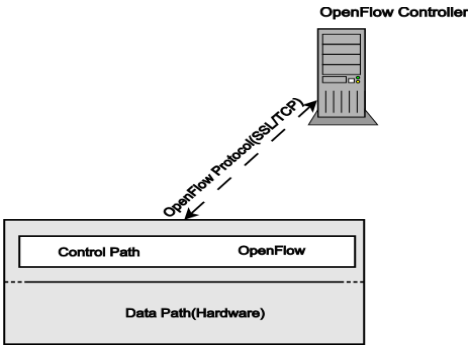

Figure 1: Open Flow Protocol.

In this paper we have proposed A optimize algorithm to find best shortest path on the basis of packet loss and delay of all possible path between sources to destination. In normal SDN routing one shortest path is selected to transfer packets but in future that path may get congested due to traffic. A new path will be selected if there are packet loss goes to beyond defined threshold.

II. RELATED WORK

Ian F. Akyildiz, Ahyoung Lee, Pu Wang,Min Luo, Wu Chou [4] proposes A new protocol CPRecovery run independently on top of network operation system in their research. That is coordination protocol between primary and backup controller. By this CPRecovery process switch send inactivity probe and amount of waiting time, if controller does not reply switch assumes that controller is failed then switch searches for next controller

and start a connection between controller and own, The controller receive join message send join event to switch and changed its state to primary controller

Monia Ghobadi, Soheil Hassas Yeganeh [5] takes the advantages of global view of SDN controller to make faster and accurate congestion and introduces new research in SDN field that is rethinking of end to end congestion control in SDN based on Transmission Control Protocol and developed Paced TCP and Open TCP tools. Proposed Open TCP dynamically changes its behavior and window size based on network traffic and used the globalization behavior SDN (like topology and routing information), relevant statics collected by SDN (like matrix and link utilization) and open modification of TCP stack by network operator.

By analyzing all above traffic engineering approach we developed a new congestion aware routing algorithm for SDN network.

III. CONGESTED LINK DISCOVERY

OpenFlow is a programmable network protocol and OpenFlow enabled switch manage and direct Network traffic according instruction coming from centralized controller. Presently traffic is characterized according to time and application behavior and requirement. Early in telephone network, TE was not efficient as it used static routing. Fixed routers always are there to optimize the busy hours, even if it not busy ours. This is very expensive option and lack of flexibility. Dynamic routing is when resource allocated at run time and periodically update network and helped flexibility. Dynamic routing can be based on time, events and stats. Flowing is the metric calculation we applied for congestion calculation.

A. QoS Metric calculation:

QoS is classified by additive and non-additive [6]. In additive links weight is addition of all weight of the link along that path. And for non-additive link QoS weight should be minimum or maximum of link along that path. Let G(E, N) denote the network topology, where E is set of links and N is set of node. We define cost *fC* and delay *fD*.

s⟶ source

d⟶ destination

Let d (i, j) be a metric for arc (i, j). For any path st = (i, j, k . . . 1, m),

$$W_k(P) = \sum(w(i, j)) \leq C_k$$

(For k = 1, 2, 3 ...*m*) all Path satisfying equation 3.5 are feasible path and best path is optimal path

$$f_C(r) = \sum c(i, j)$$

f_C is define congestion and f_D delay measure.

$$f_D(r) = \sum d(i, j)$$

B. Proposed packet loss discovery:

It collects current state information of such as link speed, packet drop and available bandwidth from the forwarder. Controller request various statistics from forwarding element by sending FEATURE REQUEST message, and forwarding element reply FEATURE REPLY message have request statistics. Packet loss measure calculated by route management that is

$$Pij = \begin{cases} (T_{ij} - B_{ij}) & B_{ij} < T_{ij} \\ 0 & B_{ij} > T_{ij} \end{cases}$$

T_{ij} Speed of packet between node i and j.

Bij Link bandwidth between node i and j.

$c_{ij} = P_{ij} + d_{ij} \forall (i, j) \in E$

Cost metric P_{ij} denote congestion measure for traffic on link (i, j), d_{ij} is delay measure. All these parameter collected by route management function.

IV. CONGESTION AWARE ROUTING(CAR)

Congestion aware routing is proposed routing which is perform after calculating and analyzing statistics of Open Flow switches attached with network. We will assume congestion will occur if there is a packet loss at any ports of switch. Congestion on network is dynamic it may change at different time. We will collect statistics of switch port at regular interval of time. Algorithm 1 is statistics collection at regular interval of time.

Algorithm 1: Time based statistics collection.

1. Send state query to every switch in regular interval;
2. Collect state reply message from every switch;
3. Process every state reply message and create statistics table for every switch;

Packet loss at port is difference of incoming traffic and outgoing traffic at that port. If there is congestion packet loss occurs.

Algorithm 2: Statistics collection request.

1. Create reference statistics request of class OFStatisticsRequest;
2. Store OF type statistics request message;
3. Create object of class OFPortStatistics Request;
4. Go to setPort Number;

Algorithm 2 is for statistics collection request from all switches present in topology. In first step controller request port statistics from all switches present in network. In second step first we access OpenFlow statistics request from OFStatistics Request class. Third step is Open Statistics request message and store it in an array. In last step controller take port number from all switches.

A. SDN topology

Mininet is a network emulator for the experiment of OpenFlow and software defined network. We used this emulator for the creation of simple network topology is used for our proposed work. It has nine switch and

all switches connected with three host. We will test by making heavy traffic from source to destination. Figure 2 is simple SDN network which have one central controller and ten OpenFlow switches all switches have three host.

B. Congested network

Figure 3 shows congestion on link associated with switch S2 due to heavy traffic, centralized controller have global view of network and temporarily block congested link for some threshold time and run Dijkstra algorithm to find new shortest route in network. Dijkstra algorithm runs regularly.

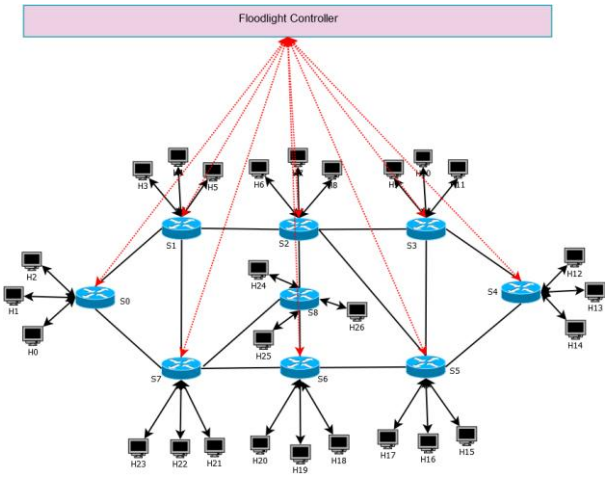

Figure 2: Topology.

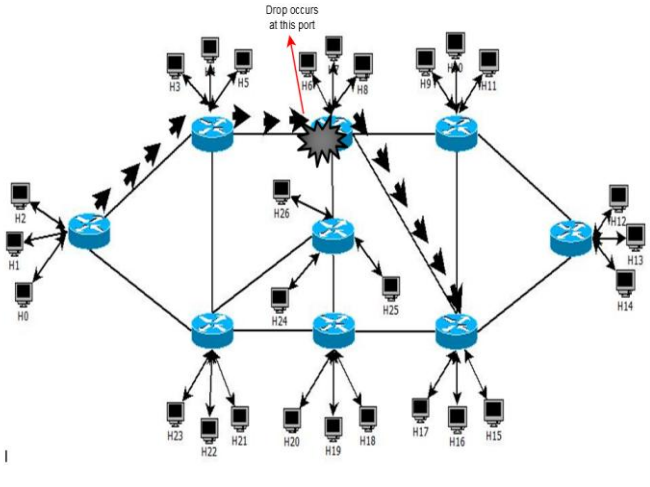

Figure 3: Congestion at switch 2.

Controller get port statistics of all switch associated with path. After getting port statistics controller calculate packet loss and reroute to next shortest path. Shortest path algorithm e.g. Dijkstra run regularly and get new shortest path to transmit packet.

Figure 4 shows new shortest path find by dijkstra algorithm. Central controller install new rule to all forwarding devices and that rule may change if network modified. Here in we used floodlight SDN controller for controlling network.

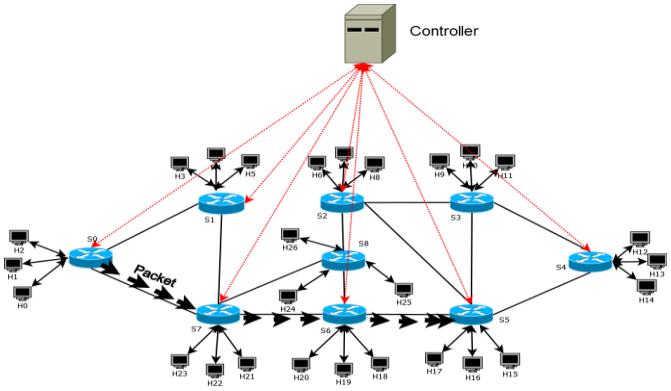

Figure 4: Route After Congestion Recover.

V. CONCLUSION

In this paper we described traffic engineering in SDN architecture. We discussed related work SDN networking. Traditional traffic engineering is done by ATM networking which is static and inefficient. Current IP and MPLS based traffic engineering used by centralized controller which have global view of entire network. SDN network is extension of current networking a centralized controller have information of all network and this behavior we used for congestion aware routing. This proposed routing is based on packet loss and delay. we reduce the packet loss in network for better performance and improving over all network performance. This marginal performance can be more efficient for the big network.

5.1 Future Work

In this paper implementation we examined number of important issue in QoS related to congestion aware routing and optimization. CARs is part of resource management. We will take multiple constraint cost metric to find best optimal path in given feasible path. We can use heuristic technique to do this optimal path search.

REFERENCES

[1] "www.opennetworking.org/sdn-resources/sdn library/technical-papers," 2014 (accessed Sptembber 14, 2014)

[2] S. J. Vaughan-Nichols, "Openflow: The next generation of the network?," Computer, no. 8, pp. 13–15, 2011.

[3] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, et al., "A survey of software-defined networking: Past, present, and future of programmable networks," Communications Surveys & Tutorials, IEEE, vol. 16, no. 3, pp. 1617–1634, 2014

[4] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in sdn-openflow networks," Computer Networks, vol. 71, pp. 1–30, 2014..

[5] M. Ghobadi, S. H. Yeganeh, and Y. Ganjali, "Rethinking end-to-end congestion control in software-defined networks," in Proceedings of the 11th ACM Workshop on Hot Topics in Networks, pp. 61–66, ACM, 2012

[6] TF Yu, K Wang, YH Hsu, "Adaptive routing for video streaming with QoS support over SDN networks," 2015 International Conference on Information Networking (ICOIN), IEEE. pp. 318 - 323.