Abstract – The increasing traffic in data centers are affecting the performance due to bottlenecks and can result in packet loss. Hence multipath routes are better choice compared to single route option. Usage of multipath to route traffic can lead to lower management cost and increased performance at data centers. Since the traffic informations remain almost similar everyday in the data centers, the implementaion of Neural network is possible inside switch/router which can compose routing between origin and destination. Multipath routing with neural network can achieve dynamic load balancing and optimized routing without intervention from the SDN controller.

Keywords – Data Center, SDN, OpenFlow, Network Neural, fat-tree topology.

1. Introduction

The traffic that flows from the data center to users through the Internet has grown at a compound annual growth rate [2] of 23 percent from 2013 to 2018. Traffic growth rate per year EB 513 and EB 1457 respectively. Data centers are the main connection between users and various services offered (e.g. internet banking transfer), independent of the the user equipment such as computer, cell phone etc. One factor that could enhance the performance of our data centers is to use multipath routing over single path routing which can be congested easily.

Software Defined Networking (SDN) introduces the separation of the control and data planes, the flow abstraction upon which forwarding decisions are made, the (logical) centralization of network control, and the ability to program the network [3].

The SDN employs OpenFlow 1.3 protocol [1] for monitoring network switches, sending policy commands to each switch, programming routes using flow tables that can define the path information. Furthermore, the protocol allows for dynamic reprogramming of network devices through an external controller that contains the control logic. The load balancing methods can be divided into two types: static load balancing and dynamic load balancing [6], in Static load balancing it is not possible to reassign the calculated route after data is transmitted. However, the dynamic load balancing method allows us to modify the router according to the continuously updated traffic statistics.

A dynamic load balancer[6] uses a performance management (PM). PM measures key indicators of network performance such as latency and packet loss for each route. This module can detect performance fluctuations in the network and if combined with an SDN controller, the controller can dynamically reconfigure the network when bandwidth fluctuations occur.

It may be the case that the information about the traffic generated in a data center is similar every day and the routing information doesn’t change. In this scenario neural networks can be used to develope control logic according to the system behavior, and can be trained. Neural network will be implemented in each network device to choose several the possible paths or routes, to send info, and to create a dynamic load balancer.

The present work aims to use multiple paths between source and destination for a data center with fat-tree topology while being controlled by the neural network implemented in each switch or router. A collection system to collect metrics about bandwidth and and packet loss of each switch or router will be used to train neural networks. Our proposal will be compared with two load balancers: (random) static load balancer and ECMP load balancer.

2. Proposal

2.1. System Architecture

The objective of this study is to route data traffic using the previously trained neural network. The
path are chosen with greater availability of bandwidth and lower packet loss. The neural network algorithm is implemented inside the Open vSwitch (OVS)[5] as an internal controller for routing traffic. As it does not depend on a central controller (SDN) for a switch or router, it eliminates processing time in the routing decision. The internal control switch or router will enhance the performance of the data center.

A general scheme is shown in Figure 1.

The protocol chosen is OpenFlow 1.3[1], where we use the Group Table function that allows to interact the neural network algorithm inside each switch or router. It is not controlled by any external controller to program the flow tables. The sFlow-RT program allows to obtain statistics and OpenDaylight sends the parameters for each neural network that is on the switch/router. A program developed in python named SeRMoP (Collection Service, monitoring, parameter setting, and sending data) will allow to collect metrics from all the switches and routers using the REST API. The different collected metrics are bandwidth availability for each output port and packet loss on each link from the possible paths. Furthermore, it conducts a real-time monitoring behavior of the traffic information. Next the parameterized entries will be sent to each switch or router to be used by the neural network in the routing decision. Every 5 min this procedure will be repeated.

2.2. Topology
To route data traffic between user and server, fat-tree topology presents multiple paths, allowing the sending of information by various routes.

The Figure 2 shows the fat-tree topology data center. It consists of three levels, where the first level is core, second level is aggregation and third level edge. Each node represented as a switch or router will have an implementation of a backpropagation neural network. At present there are no research proposal work in reference to our proposal. Our simulation uses Mininet network simulator[4]. We implemented the OpenFlow switch with Open vSwitch (OVS)[5].

2.3. Neural Network
We used the back propagation neural network (BPNN). We had 4 input nodes, 3 hidden nodes, and 2 output node as shown in Figure 3. The inputs are the availability of bandwidth and packet loss, for example in figure 3 shows the neural network to the switch 5 is:

![Figure 3. BPNN 4 input, 3 hidden, 2 output](image-url)
As is connected to SW2 and SW3, inputs for BPNN are bandwidth (BW): Port 5-2 and Port 5-3, also packet loss (PL): Link 5-2 and Link 5-3.

The collected data (available bandwidth and packet loss) is responsible for parameterized BW and PL in the range of 0 to 1, to send the data to the neural network, for example the value 0.5 representing 50% of the available bandwidth or the value 0.1 representing 10% of availability of bandwidth.

The neural networks are used to be trained of the specific job like multipath routing while utilizing most of the infrastructure link data and achieves dynamic load balancing depending on the bandwidth and packet loss.

2.4. Load Balancing Method
The use of a predetermined traffic in the static load balancer allow knowing the behavior of BW and PL on switches with other switches. These data will be collected and used to train neural networks. The same experiment was developed with the load balancer ECMP. Then the path or multipath will be defined by the training of BW and PL with respect to the communication between neighbor switches.

The behavior of the neural network to decide where to send packets, will be illustrated in figure 4, for example, if the PC connected to switch 5 wants to communicate with the PC of the switch 10, possible routes are:

- 5-3-1-8-10
- 5-3-1-8-9-7-10
- 5-3-4-2-6-7-10
- 5-3-4-2-6-7-9-8-10
- 5-2-6-7-10
- 5-2-6-7-9-8-10
- 5-2-4-3-1-8-10
- 5-2-4-3-1-8-9-7-10

When a packet reaches the switch 5, the BPNN decides to send the switch SW3 or SW2. If the switch 2 is elected, this will be evaluated again by the BPNN (Switch 2) on each route. The BPNN decides where the data will go. It may be the case where it divides, the data flow for both routes, such as 50% for each route, or also send 70% in the first route and the other 30%.

For this example we illustrate the behavior BW for the static load balancer:

![Figure 4. Fat-tree routing](image)

Figure 5 shows that the BW is reduced because of using the same route between the origin and destination.

The behavior BW for the ECMP load balancer:

![Figure 5. Fat-tree routing](image)

Figure 6 shows that the BW increase compared to Figure 5 and has instants where more increases BW.

3. Conclusion
Neural networks can develop new routes or multiple routes, in the data center for data traffic. The benefit
of having multiple paths between source and destination allows for greater bandwidth with less packet loss. The OpenFlow protocol allows implement algorithms inside the switch as an internal controller.

References


