**Improving Software Defined Wi-Fi Network Performance by Optimising Load Balancing Technique**

**Introduction:**

Traditional network which has been in practise for years have implemented the network functionality directly in the appliance. That is the data plane and the control plane are implemented on the appliances. The appliance could be any networking device like router or switches. Most of the functionality of these networking device are implemented on their hardware. This makes the configuration to be very time consuming. Moreover, since it is done manually on all the systems, there are more chances for the error to occur while configuring it. Apart from this, if the organization has multiple devices from multiple vendors, it further increases the complexity in configuring it. To overcome all these drawbacks, Software Defined Networks were introduced. The core of the software defined network is to decouple the hardware and software from the networking device. That is the data plane which is responsible for forwarding the traffic from source to destination is left to be on the networking device and the control plane which is responsible for determining where the traffic should be sent is moved out of the hardware. Refer Figure 1 at the end of the document. This abstraction of the control plane from the hardware gives way for having a centralized software application containing the control planes of all the connected networking devices. The advantage of having a centralized control plane is that it increases the configuration accuracy, flexibility and consistency. Having a centralized control plane makes it easier for the network administrators to update all the system at once, increasing the accuracy and making it less error prone. This approach also helps us to implement our own logic for the control plane instead of depending on the vendors algorithms. In this paper we will be referring few important IEEE papers on how to improve the performance of the Software Defined Wi-Fi Networks by implementing an efficient load balancing algorithm. This paper will cover the most efficient proven techniques on how the load balancer should be implemented in order to improve the performance in Software Defined Wi-Fi Networks. The working of the SDN is shown in figure 2 at the end of the document.

**Strategies on how to improve the SDN Wi-Fi Performance:**

Load Balancing is a technique of distributing the load over the servers based on the number of clients connected to the server in order to increase the capacity and reliability of applications. This process also decreases the burden on the server. Below are the different techniques used improve the performance of the Software Defined Wi-Fi Networks.

1. Using of dynamic algorithm when compared with the static algorithm can reduce the average response time [1].
2. Considering the number of active client connections and using the concept of shortest path between the server and client, distributes the clients evenly while avoiding the congestion [2].
3. Reducing the load of the Access Points by reducing the transmission power of the Access Points helps in reducing the jitter when there is a handover between the stations and reducing the packet loss [3].
4. Accumulatively considering the Access Points side information and the Station side information helps in achieving good distribution of load [4].

The below paper is organised as follows. All the above strategies are explained in detail on how it is achieved and how it is evaluated. Following it will be section explaining a suggested idea based on the above strategies which could be implemented on Software Defined Wi-Fi Networks to improve the performance.

**Dynamic Algorithm vs Static Algorithm for Load Balancing:**

As explained in the above, Traditional networking faces lot of drawbacks when compared to the Software defined networking. Network load balancers plays a very vital role in providing a good performance as they are the entry points to the network and their main job is to efficiently spread the load across the devices. Traditional load balancing techniques might not be as efficient as the Software defined load balancing techniques. This is because in case of the Software defined networking, it can have a global view of all the connected networks. This give it an upper hand in having more useful information to determine how the clients should be distributed among the servers. Moreover, in case of Traditional networking, the load balancers are implemented on the hardware. This makes it inflexible as they are proprietary and closed. Also, this doesn’t support overwriting the load balancing algorithm given by the vendor. In contrast to this SDN which has the centralized software for all the connected devices, supports writing custom algorithms. The load balancing algorithm can be of two types – Static and Dynamic. In case of the static algorithms, information which are local such as Random or Round Robin are taken into account. This information doesn’t consider bottleneck conditions. Whereas in dynamic algorithms, they have a global view of all the connected devices, hence can make use of stateful information while deciding on how to distribute the load across the servers. In case of the SDN network, there is an overhead because of the communication between the centralized control plane and the data plane. This is because the information has to be passed from the control plane which is located outside the hardware to the data plane. Due to this we have two different approaches for an SDN network – Reactive and Active SDN load balancers. In case of the Reactive SDN load balancer, all the control logics are taken care by the SDN controller. Whereas in the Active SDN load balancer, though the control logic is handled by the SDN controller, few of it is still left to the switches or the routers to take care. Thus, making it a bit smarter than the ones in the Reactive SDN.

Performance evaluation is done for the SDN load balancers for performance evaluation of both the active and reactive SDN load balancers and also with different scheduling strategies. The first hypothesis is to have a single web server. This web server is configured in such a way that it produces less average response time than the reactive SDN loan balancer. This hypothesis is to measure the impact of the communication overhead between the controller and the data plane. The result of this hypothesis is that the average response time by the reactive SDN load balancer is higher when compared with the web server. This can be used as an evidence to conclude that we cannot reject the fact there is an overhead because of the communication between the centralized controller and the data plane in the networking device. In the second hypothesis, we try to prove that the dynamic algorithm is better than the static algorithm in a reactive SDN load balancer. We make use of CPU usage statistics for the dynamic algorithm and in case of static algorithm we make use of Round Robin and Random strategies. Both real and virtual scenarios were used to test this. The result of this hypothesis is that the dynamic algorithms are better than the static algorithms but not in all circumstances. Because the results were evidential only when there were huge number of requests and in the virtual environment. The last hypothesis was to evaluate the performance of both the active and reactive SDN load balancer against each other. The result of this hypothesis is that the reactive SDN load balancer was better than the active SDN load balancer. Though, in the beginning the active SDN load balancer seemed to have lesser response time, once the number of request increased, it turned out that the reactive SDN load balancer is better than the active SDN load balancer.

This works concludes that the reactive SDN load balancer does a more efficient job in distributing the load across the servers, even though there were communication overhead in them. It also concluded that the dynamic algorithm is better than static algorithm while taking into the factors for distributing the load over the servers.

**Number of active connections and Shortest Path – Strategy:**

The strategy which is being implemented here is to consider two parameters in designing an algorithm for load balancing in Software Defined Networking. The parameters are the number of connections which are active to an individual server and the shortest path which is calculated between the client and the server. The working of this algorithm is explained in the below paragraphs. Whenever the client requests for something, the load balancer comes into picture and decides the best server amongst all the connected server to serve the request for the client. The load balancer considers the number of active connections in each of the server as the parameter and decides the server with least connection number as the best to serve the request.

Two concepts which are implemented in this strategy are explained below. The first one being the number of active connections. The load balancers job is to route the incoming request to the server with the least load. There are many ways of measuring the load of the server. One such measuring method is to consider the number of active connections. The load balancer first collects the information about the network topology. That is, it computes the number of active connections per server and how the servers are interrelated with each other. This information is acquired using RESTCONF Northbound API. Once the load balancer has this information in hand, it is ready to serve the incoming requests. On receiving a request, the load balancer will direct the request to the server with least number of active connections. The load balancer also keeps track of the requests which are being currently processed by the server. By doing so it makes sure that the strategy of load balancing considers the current load conditions. The second concept is to make use of the shortest path between the server and the client. In order to find the shortest path, Dijkstra’s algorithm is being used. This algorithm is used her so as to find the shortest path between the client and server which has been selected based on the least connection. The controller provides the network statistics such are receiving rate and transmission rate which could be used to find the cost for each edge. The flow rules are updated accordingly by the load balancer. Also, the network statistics are updated every minute, so that the Dijkstra’s algorithm gives accurate results.

The implementation of this strategy is explained below. The load balancer’s first job is to scan through all the connected devices and gather all the information such as the topology, number of connections, devices etc. This information is gathered using RESTCONF API. The load balancer then finds the server with the lease number of active connection. After which it finds all the possible paths between the client and the server. Then using the Dijkstra’s algorithm, it finds the shortest path between the client and the server. This is possible by making use of the statistics such as receiving rate and transmission rate from the controller. Then this path information is sent to the switch flow table. The load balancer keeps checking the server’s connection status, computes the latest server with the least load, computes the latest available congestion free path. The load balancer updates the value dynamically making this a dynamic algorithm.

This strategy of using two concepts – number of active connections and shortest path is more efficient than the strategy which uses random load balancing. The idea of using the least number of active connections to find the server with the least load and the idea of using the Dijkstra’s algorithm to find the shortest path between the client and the server helps us in designing an algorithm which avoid the congestion and also distribute the load evenly across all the servers based on the loads of the server.

**Reducing the coverage cell – Strategy:**

This strategy of reducing the coverage cell helps in avoiding the overloading and prohibiting the congestion. WLAN is one of the most popular wireless networks. This is very famous because of many advantages such as cheaper cost for the installation and very high-speed communication speed. The working of the mobile stations in a WLAN network is as follows. The mobile stations (STA) connects to the Access Point (AP) which has the highest signal strength. The issue with this is that it increases load on that server even though there were enough servers to handle the mobile stations equally. Network congestions and overloading on the servers happens due to unbalanced loads. Load balancing techniques are required to solve this. Using the load balancer helps in increasing the throughput, avoiding the overloading, maximizing the utilization etc. Bringing the concept of software defined network to Wi-Fi, opens up lot of advantages such as automation, having a centralized controller for all the APs. It also allows programming capabilities on the controllers.

The mobile station usually connects to an access point based on the signal strength. That is, the access point with the strongest Received Signal Strength Indicator (RSSI) will be chosen by the station. As discussed above the issue with this is that, there are more chances that most of the stations gets connected to the same access point even when there are other access points with lesser load. This causes imbalance. Due to this imbalance the quality of service (QoS) which is provided to the stations are diminished. The access points can be of one of the following states – Balanced, Underloaded and Overloaded. As the name suggests, overloaded access points are the ones which are currently serving more stations than the average, underloaded access points are the ones which are currently serving stations less than the average or not serving any and balanced access points are the ones which are serving almost the average capacity of the mobile stations. The idea proposed here is that the coverage cell of the overloaded access points can be reduced in order to forcefully make the stations to connect to nearby access points. The point to be noted here is that the access point to which the mobile station is forcefully connected should be an underloaded access point or the balanced access point. The coverage cell can be reduced by reducing the transmission power of the access point.

The use of SDN in WLAN has made it more flexible in terms of installing network polices, increasing the scalability and making it more agile. Having a centralized controller helps in monitoring the flow of the network and also can direct how the network should flow in that network. The technique used here is dynamic in nature. The idea being forcefully making the mobile stations which are in the overlapping regions to hand off to the nearby access point which are under loaded or balanced. There are in total three phases which are explained below. The first phase is the calculation of the load of the access point. The access point can either be underloaded or overloaded. Underloaded is where it is serving less than its actual capacity and overloaded is where it is serving more number of stations and it cannot take any more requests from the mobile stations. Two parameters are used to calculate the load of the access point. Those parameters are the number of mobile stations which are connected to the access point and the second parameter is the channel’s throughput value. This is the end of first phase. Second phase is where the load of the access point is monitored. The SDN controller continuously monitors the state of all the access points. Whenever the access point state changes, the SDN controller acts accordingly. That is, if the access point’s state changes from overloaded to underloaded, then the SDN controller registers its new state and takes no further action. But if the access point’s change from underloaded to overloaded, then the SDN controller registers its new state and also it sends the rule to the access point to perform a forced handover of the mobile station. The third and last phase is the forced handover execution. The access point receives the new rule from the SDN controller which would ease up the overloaded access point. The overloaded access point, then force the mobile station to handover to the target access point. This is done by sending a message to the target access point requesting for the RSSI. If this value is greater than the threshold, then the home access point requests the target access point to delete the mobile station entry and execute the handover. This is because when the RSSI value is greater than the threshold, it means that the access point is already overloaded and cannot accept new connections. The home access point continues to execute the forced handover until it loses few of the stations and become under loaded and parallelly the mobile station continues to associate with the target access point.

The experiments performed using this load balancing technique proves that the average packet loss occurred using this load balancing strategy is less when compared with the which doesn’t use this technique. Additionally, it also proves that the jitter values are smaller when the new load balancing technique were implemented when compared with the one which doesn’t use this technique. Hence, we can conclude that this strategy of reducing the coverage cell to reduce the load on the servers helps in efficiently reducing the packet loss and jitters in the mobile stations.

**Using AP side and STA side information – Strategy:**

The last strategy which should significantly improve the performance in software defined Wi-Fi networking is to make use of information from both the access point side and also from the mobile station side. We can see that the above strategies and most of the strategies considered in designing an algorithm for load balancing uses only the information from the access point side but not from the mobile station side. Hence the strategy here is to consider both this information together to design an algorithm. The south bound interface in the software defined networks initially supported only the switches. They didn’t have any support for the wireless networks. But this has been changed over the years. OpenFlow (south bound interface) has taken different versions like OpenRoads and Odin which support the wireless networks. The problem statement for this strategy is same as that of the previous strategy. When the RSSI value of the access point is too high, all the mobile stations tend to connect to that particular access point even when there are other access points which can serve these mobile stations. Due to this high load on one access point it results in poor utilization of the network. Hence load balancing techniques are used to solve this unbalanced distribution of loads in the network. But all the load balancing techniques which are proposed for solving this uses only access point side information for finding an efficient access point amongst all the access points to serve the mobile station. Our strategy is to make use of the network controller which has the global view of the networks and then make use of both the access point side information and the station side information to design a load balancing algorithm for high density Wi-Fi networks.

There are many metrics to calculate the load on the access point. One such metric which is used in this strategy is the measure of number of active mobile stations to the access point. The algorithm used here is called Adaptive Connection and Hand-off (ACH). There are two important parameters used in this algorithm – MAXLOAD and MAXDIFF. MAXLOAD is the maximum threshold value of the access point. MAXDIFF is the maximum difference value between the loads of the access points. Once the load value of the access point is greater than the MAXLOAD. Adaptive Connection and Hand-off algorithm is triggered. Also, when there is a difference value between the access points, this Adaptive Connection and Hand-off algorithm is triggered. This is expected because, when there is a difference in load it means that the loads are not distributed evenly in the network.

The algorithm used here is explained below. The first step of the network controller is to collect the information of the mobile stations and the access points. Then for each of the mobile stations, the following set of activities are performed. For each mobile stations the set of access points which could be sensed are listed down. For all the listed access points, choose the access point which has the least number of stations associated with it. Then that respective mobile station is associated with that access point. The network parameters such as load max and load min are updated. If this value of load max is greater than MAXLOAD or if the difference in load max and load min is greater than the MAXDIFF, then find the access point which has the highest load. Then pick up the station which has the weakest mobile station associated with that access point. Then find an access point which is underloaded and then hand off the weakest mobile station to that underloaded access point.

The findings of this strategy are as follows. The load has been uniformly distributed by using Adaptive Connection and Handoff algorithm when compared to other popular used schemes. By ignoring the access points with the weakest RSSI, the throughput achieved using this algorithm is found to be better. Hence with this Adaptive Connection and Handoff algorithm, load balancing could be better achieved and there is a good trade-off between the fairness and the throughput in the network.

**Suggested Strategy:**

As we see from above, there are many different strategies which are deployed to make the load balancing efficient in the Software Defined Wi-Fi Networking. Thus, improving their performance. In this paper we are suggesting a strategy which combines the proven results of all the above discussed strategies in improving the performance of the Software Defined Wi-Fi Networks. The two main strategies which are considered here are the one which reduces the coverage cell and the other one considering the information from both the access point side and the mobile station side. The other two strategies like use of dynamic algorithms and making use of the number of client as the metric to calculate the load is incorporated in our strategy. Hence, they are not stressed on this point. The strategy is explained in the below paragraph.

Our idea is to use a dynamic algorithm. That is, the algorithm will be based on dynamic data such as number of active connections, signal strength of the stations etc. This helps in constructing an efficient load balancing algorithm. For calculating the load on the servers, we make use of the count of the number of active connections / mobile stations. The third strategy which is about reducing the coverage cell, though being more efficient, has few draw backs. This strategy focuses on the mobile stations which are in the overlapping regions of the access points for the handoff to underloaded access points. This doesn’t consider the mobile station which are not in the overlapping region and are in a weaker range but still in connection with the access point. If the algorithm tries to reduce the size of the coverage cell by reducing the transmission power, the mobile stations in the overlapping region might handoff the underloaded access point but not the ones which are in connection but at a farther range of that access point. When the range of the coverage cell reduces, the connection between the access point and the mobile network would terminate. In order to solve this issue, we should make use of information from the mobile stations in determining whether to reduce the coverage cell or not. The idea of use of mobile station side information along with the access point side information is acquired from the fourth strategy discussed in this paper. Our strategy is to combine the implementation of Adaptive Connection and Hand-off algorithm with the above strategy. The pseudocode is as follows

Network controller collects the access point side information and mobile station information

* For each of the mobile stations
  + - All the sensed access points are listed by that mobile station
    - Find the access point which has the least connections
    - Connect the mobile station to the above access point
    - If (load max > MAXLOAD) or ((load max – load min) > MAXDIFF) then
      * + Select the access point with the maximum number of connections
        + Pick the mobile station which has the weakest strength

If the mobile station is in the overlapping region of the access points

Reduce the coverage cell region by reducing the transmission power

Else handoff the mobile station to the nearby underloaded access point

* + - Else
      * + Continue
    - End if
* End for

Implementing the above algorithm helps in avoiding the disconnection of the mobile station from the access points which are far from the access points due to the reduction in the coverage cell.

**Conclusion:**

This paper summarizes various techniques used to improve the performance of the Software Defined Wi-Fi Networking by tweaking the load balancing algorithm. After the summarization of all the strategies, we have also suggested an idea of how to improve one of the already proven strategy by coming the ideas from other strategies.

**References:**

[1] – “A Performance Evaluation of Software Defined Networking Load Balancers Implementation” by Walber Jos´e Adriano Silva, Kelvin Lopes Dias and Djamel Fawzi Hadj Sadok – 2017 ICOIN Conference

[2] – “Dynamic Load Balancing Strategy in Software-Defined Networking” by Saket Bhelekar, Mrdvika Iyer, Gargee Mehta and Sheetal Chaudhari – 2017 ICEI Conference

[3] – “Dynamic Load Balancing Technique for Software Defined Wi-Fi Networks” by Ahmed S. AbdelRahman and Ashraf B. El- Sisi – 2017 IEEE Conference

[4] – “Achieving Load Balancing in High Density Software Defined Wi-Fi Networks” by Ze Chan, Sohaib Manzoor, Yayu Gao and Xiaojun Hei – 2017 ICFIT Conference

[5] – “Dynamic Load-Balancing Mechanism for Software-Defined Networking” by Wen-Hwa Liao and Ssu-Chi Kuai and Cheng-Hsiu Lu – 2016 ICNNA Conference

**Figures:**

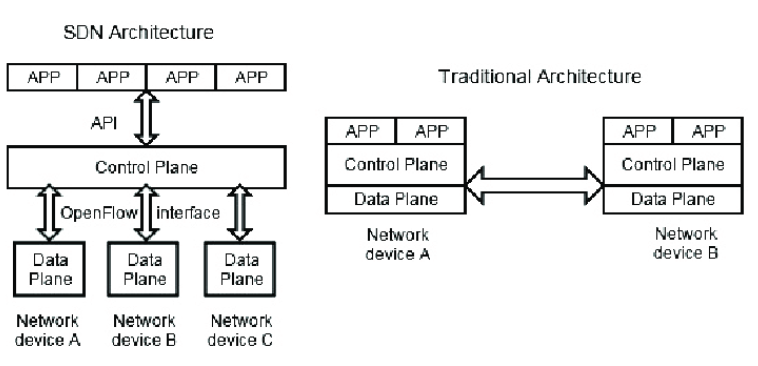


Figure 1 – Difference between SDN and Traditional Network

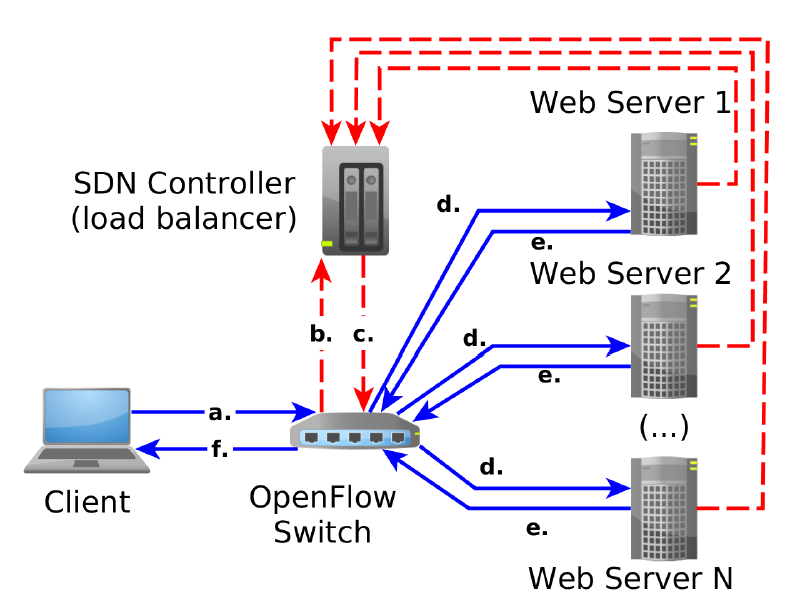


Figure 2 – Load Balancing in SDN