Algorithms for optimizing packet propagation latency in software-defined networks

I G Ivanov¹, G V Hristov² and V D Stoykova¹

¹Trakia University, Faculty of Technics and Technologies, 8600 Yambol, 38 Graf Ignatiev str., Bulgaria
²“Angel Kanchev” Univresity of Ruse, Bulgaria

e-mail: ivan.ivanov@uni-sz.bg

Abstract: With the development of technology in recent years, it is clear that the potential in traditional networks has been exhausted. Software defined network (SDN) is emerging network, which provides a global centralized view, separating the control plane from the data plane. This article discusses the potential for determining the optimal latency in SDN networks with capital cost optimization – CapEx. A mathematical formulation for solving the problem is considered. Following this approach will speed up routing and improve QoS.

1. Introduction

Architectural concept for software-defined networks (SDN), in which the management of traffic in a computer network is separated by a mechanism for distribution and processing of data packets through an application programming interface (API), including traffic analysis and ensuring its operation, so and optimizing packet propagation delay time. Delay time is the period between each user's action on the network and the web application's response to that action, which in network terms is often referred to as "total time for a two-way packet path".

Later in this article, instead of "delay time", the shorter term "latency" will be used for the total average time during which the two-way processes of each packet go unnoticed. Latency is usually measured in milliseconds (ms) and is unavoidable due to the way networks communicate with each other. The main components that affect the overall latency of the network are:

- Transmission environment. This is the physical path between the starting and ending point. The type of environment can affect latency. For example, older networks based on copper cable have a higher latency than modern optical fibers.
- Distribution latency. The farther away the two nodes of the network, the greater the latency, as it depends on the distance between the two communicating nodes. Theoretically, the latency of the packet carried in a circle around the world (two-way) is 133 ms. Such a two-way path takes more time, although latency decreases when direct connections are made through the network backbone. The latency of packet distribution and delivery greatly affects SDN performance, especially when intra-band communication is used.
- Latency of queue waiting. This is the time the packets wait to be sent.
- Process latency. This is the time it takes routers to review the headers of packets and determine where to target them.

There is a consensus that propagation latency usually dominates WANs and is primarily a function of the location of the source and the destination of the packets. In the context of SDN, this means that...
the location of the controller in terms of SDN network switches and / or routers is directly proportional to the latency of propagation. Therefore, the location of the controller is an important issue in the design and implementation of SDN where guaranteed quality of service (QoS) is desired.

Typically, long-range networks are segmented into several smaller administrative domains, each controlled by a dedicated controller. This is necessary to facilitate network scalability, gradual deployment and respond to potential security threats. Thus, for a given network topology, it is important to determine the optimal number of controllers to use so that overall latency is minimized while maintaining a fair load distribution between the controllers.

At the same time as minimizing overall latency, network operators are particularly sensitive to saving capital costs - CapEx, associated with building networks. Therefore, it makes sense to take into account the cost of installing controllers for a given topology by determining their optimal number. For this reason, in order to minimize the trade-off between performance and costs, it is important to determine the optimal overall network latency.

In this sense, the main objectives of the development described in this article are formulated, namely:

- Use of algorithms for determining the optimal latency of packet propagation in SDN networks;
- Determining the optimal latency of packet propagation;
- Determining the costs for installing controllers in SDN networks;
- Simulation of the used algorithms in the software environment for numerical analysis of "MATLAB" with a set of data on the network topology of Evolink Bulgaria.

This article discusses algorithms for determining the optimal latency of packet propagation and the cost of installing controllers to be used for a given topology. Optimization depends on the average packet latency for a given topology, the number of SDN deployment controllers, the algorithms that have been used to locate, and the number of controllers to achieve high quality of service (QoS). The considered algorithms were tested in the software environment for numerical analysis "MATLAB" with a set of data on the network topology of Evolink Bulgaria [1].

2. Material and methods

High-bandwidth computer networks (WANs) are usually divided into smaller administrative domains to provide failure protection and load balancing in the control plane.

SDN controllers are primarily responsible for providing services such as: Flow records to switches, Load balancing, Detection of threats and error recovery in the connection layer. Other indicators that also matter are Reliability and Performance.

The article deals mainly with long-range networks - WAN, where latency of distribution dominates and is an important parameter of QoS for network services. The problem that arises is: To find the optimal average latency of distribution in the network topology of Evolink Bulgaria with optimal benefits of capital costs - CapEx.

It is known that the location of the controller in local area networks (LAN) is relatively simple. In a WAN, however, the location of the controller is a combinatorial optimization task that cannot be solved in polynomial time. In summary, it is essential to determine the optimal number of controllers to use for a given WAN topology. This is because a controller can provide suboptimal performance in terms of packet propagation latency, load balancing, scalability, and security.

Therefore, the problem to be solved is the following: given the WAN with SDN activated, how many controllers are needed to optimize packet propagation latency by focusing on two QoS parameters, namely:

- The average latency of packet propagation;
- Worst case of packet propagation latency.

Some authors have tried to minimize the mean latency of propagation, called the k-median or k-mean at the worst of the propagation delay, called the k-center problem. The authors find that both latencies cannot be minimized simultaneously [2].

In case of already optimized latency, to determine optimal capital costs for the installation of the controllers.
In the mathematical model, the network topology (or WAN) is modeled as an undirected graph $G (V, E, L)$, where:

- nodes of the graph are denoted by $V$ (network objects);
- arcs of the graph are denoted by $E$ (fiber optic connections);
- $L$ indicates the GPS locations of the network switches in degrees. For the mathematical model, $L_{avg}$ represents the average packet latency, $d(v, z')$ and is the shortest distance from a network switch (node $v \in V$) to the controller (node $Z'$), and the number of nodes is $N = |V|$ [3].

$$L_{avg} (Z') = \frac{1}{(2 \times 10^8)N} \sum_{v \in V} \min(d(v, z'))$$ (1)

The average packet latency for nodes $Z'$ is $L_{avg} (Z')$.

An alternative mean latency metric is the worst case of packet propagation for nodes $Z'$, defined as the maximum packet latency for nodes $Z'$ is $L_{wc} (Z')$.

$$L_{wc} (Z') = (\max_{v \in V} \min(d(v, s))_{z \in Z'}) \frac{1}{2 \times 10^8}$$ (2)

The notation of the average latency in the formula – $L_{wc}$ cannot be used in the simulation program with Matlab, then for this purpose the notation is entered in it: $L_{\text{worst}}$.

In this case, the problem of optimizing packet propagation latency is solved by finding the location of the set of all placements so that either $L_{avg}$ or $L_{\text{worst}}$ is minimized.

The developed mathematical model is based on the following assumptions:

- All network switches have the ability to run a software-based SDN controller;
- It is assumed that the communication between network switches and controllers takes place in the same range, control and regular traffic share the same physical connections;
- The frequency band for all optical fibers is constant;
- The security of the control road is completely solved;
- The controller and the network switches are located together;
- Documentation exists for all WAN network switch locations.

The algorithms used to solve the optimization of packet propagation latency are the following:

2.1. Partition Around Medoids (PAM)

One of the definitions of "medoid" is "an object with a data set or a cluster with a data set whose average discrepancy with all objects in the cluster is minimal".

The location of a control object in the WAN's SDN is a point of failure or attack and affects the scalability of the network. As a result, WANs are usually divided into smaller administrative domains, each of which is managed by a dedicated controller. However, it is extremely important to ensure the correct placement of these controllers to ensure QoS, which in our case is the latency of propagation.

The k-medoid method is one of the separation methods. Because it uses the most centrally located object (medoids) in the cluster to be the center of the cluster, instead of assuming the average of the objects in the cluster, it is less sensitive to noise and outlines than the k-environment approach. PAM works effectively for small data sets (for example, 100 objects in 5 clusters), but is not as effective for large data sets.

The method should be more suitable for the purposes of spatial clustering than the k-means method, due to the better quality of clustering it can achieve. However, it is well known that the k-medoid method takes a long time [4].

The mathematical model used the Partition Around Medoids (PAM) clustering solution [3] to split the network while ensuring minimal propagation latency (worst case and average latency). Unlike
classical separation methods, such as k-mean grouping, PAM is more stable in the presence of noise and extreme values, while k-mean is extremely sensitive to deviations and other extreme values.

Clustering, as shown in Algorithm 1, assumes $G(V, E)$, network graphics with switching geographic locations (longitude and latitude), $d$ a custom distance function, and $k$ the number of clusters that is analogous to the number of controllers. The initial step is to select any location of the controller. The next step is to connect each network switch to the nearest controller location using the custom distance function. For each location of the controller $l$ and switch $v$ connected to $l$, $v$ and $l$ are alternated. The average difference $TC_{vl}$ of the switch $v$ to all switches connected to $l$ is then calculated. Finally, the point with the lowest dissimilarity was chosen as the best location. The result of this algorithm is the cluster indices of each observation, the geographical location of the controllers and the distance from each switch to the controller in its area. The complexity of the calculations of this algorithm is $O(k(n - k)2)$, where $n$ is the number of switches and $k$ is the number of controllers.

Algorithm 1 describes the steps that are followed to calculate the optimal location of the SDN controllers. This approach involves the joint placement of controllers and switches. First, $k$ random switches (where $k$ is the number of controllers to be placed) are selected as potential controller locations. This is followed by connecting each switch to the nearest controller. As configuration costs (total propagation latency) decrease, the location of controllers and switches swap.

The PAM algorithm is shown in Figure 1.

| 1. Input: | $G(V,E,L)$ network graph with switch locations |
| 2. Input: | $d,k$ distance function and number of controllers |
| 3. Select $k$ representative switches arbitrarily |
| 4. for each pair of non-selected switch $v$ and selected switch $l$, calculate the total swapping cost $TC_{vl}$ |
| 5. for each pair of $v$ and $i$, if $TC_{vi} < 0$, $i$ is replaced by $v$ |
| 6. then assign each non-selected object to the most similar representative object |
| 7. Repeat steps 4 - 6 until there is no change |
| 8. Output: | $idx, CL, sumd, d$ cluster indices of each observation, controller locations, within cluster sums, and distance from each switch to controller |

Figure 1. Algorithm 1: PAM clustering.

2.2. Johnson's algorithm
To determine the best places to place SDN controllers in a WAN, you need to know the shortest paths between each pair of switches. Johnson's algorithm [5] provides a tool for finding the shortest paths between pairs of nodes and has become a popular method for solving SDN optimization problems.

This algorithm is used to calculate the shortest distance matrix used by PAM for optimal controller placement. Johnson's algorithm is a well-known optimization algorithm for calculating the shortest path between all pairs of nodes in a network. Johnson's algorithm is implemented in three main steps.

First, a node with an artificial source $q$ with zero weights is added to the network graph $G(V, E)$ to obtain a modified graph $G'$.

The Bellman-Ford algorithm is then executed for a graph $G$ with output node $q$ to find all the shortest paths $h(v)$ from $q$ to each node $v$.

If this step detects a negative weight cycle, the Bellman-Ford algorithm terminates. The edges of the original graph $G(V, E)$ are then recalculated using the output of the Bellman-Ford algorithm [6].

Finally, $q$ is removed and the Dijkstra algorithm is executed to calculate the shortest paths from each node to each other vertex in the weighted graph.

The total time complexity is:

$$O((V^2 \log V + VE))$$

Johnson's algorithm is shown in text format in Figure 2.
1. Data: \( G = (V,E) \)

2. Result: Shortest path

Compute \( G' \) where \( V[G'] = V[G] \cup s \);
\[ E[G'] = E[G] \cup (s,v) : v \in V[G] \; \]

3. for all \( v \in V[G] \) do
\[ w(s,v) = 0 \; \]
end

4. if Bellman-Ford \((G',w,s) = FALSE\) then
print the input graph contains a negative weight cycles ;
end

5. for each vertex \( v \in V[G'] \) do
set \( h(v) \) to the value of \( \delta(s,v) \) computed by the Bellman-Ford algorithm ;
end

6. for each edge \( (u,v) \in E[G'] \) do
\[ w(u,v) \leftarrow w(u,v) + h(u) - h(v) \; \]

7. for each vertex \( u \in V[G] \) do
run Dijkstra \((G,w,u)\) to compute \( \delta(s,v) \) for all \( v \in V[G] \);
end

8. for each vertex \( v \in V[G] \) do
\[ d(u,v) \leftarrow \delta(s,v) + h(v) - h(u) \]
end
end

9. Output: \( dist \), shortest path matri

---

**Figure 2.** Algorithm 2: Johnson’s algorithm.

2.3. Distance to Haversine

Given that the locations of the switches are given in GPS coordinates (longitude and latitude), calculations are performed ignoring ellipsoidal effects, which is accurate enough for most purposes.

The Haversine formula is used to determine the distance between network switches. The distance of the great circle is the shortest distance between two points on the surface of a sphere, measured on the surface of the sphere as opposed to the usual Euclidean distance.

An alternative method for calculating geographical distances is the Cosine Law. This method is optimal for shorter distances and is not good for longer distances. To calculate the distance of the great circle, the equation below is used, which equation defines the Harvesine approach [3], where:

- \( \varphi_1 \) and \( \varphi_2 \) denote the latitudes of points 1 and 2 respectively;
- \( \lambda_1 \) and \( \lambda_2 \) denote the lengths of points 1 and 2 respectively;
- \( r \) is the radius of the earth, a constant equal to 6371 km.

The optimization algorithms were applied to the real network topology of Evolink Bulgaria. The dataset for this network topology was downloaded from Topology Zoo [1], which is a database of network topologies. The key factor in the mathematical model is the distance, while the frequency band is constant. Therefore, at constant bandwidth, the propagation latency is directly proportional to the distance.

\[
2. r. \arcsin \left( \sqrt{\sin^2 \varphi_2 - \frac{\varphi_1}{2} \cos(\varphi_1) \cos(\varphi_2) \sin^2 \frac{\lambda_2 - \lambda_1}{2}} \right)
\]  

2.4. Block diagram of the optimization procedure

The diagram shown in Figure 4 summarizes the steps in deciding to place a controller. Used to model the network topology as a non-directional graph \( G (V, E) \), where \( V \) stands for network switches and \( E \)
represents the connections connecting the switches. This is followed by retrieving the geographic location data using the input dataset. To determine the weights of the edges, an adjacency matrix is applied between all connected switches.

As shown in the block diagram of Figure 3, the following is performed:

- To determine the best controller positions, first calculate the distance matrix using the Haversine formula.
- The next step involves generating the weights of the edges of the graph by implementing the adjacency matrix between all pairs of nodes.
- In the third step, Johnson's algorithm is used to generate a matrix with the shortest path for network graphics.
- Finally, the PAM algorithm was used to determine the best placements that minimize average latency and worst latency.

![Figure 3. Block diagram: Description of the model.](image)
2.5. Model simulation
MATLAB was used to simulate the model and formulations shown [7]. This model is designed to provide a tool to support Internet Service Providers (ISPs) who wish to switch to SDN so that they can optimize network performance by calculating optimal controller locations during SDN planning. Our assessment of network performance is based on distribution latency. If the network latency is taken into account, a range of maximum allowable delays must be determined. Similar Matlab studies for the analysis of the maximum allowable delay and the number of controllers are defined in [8].

Figure 4 shows the dependence of the latency on the number of controllers, calculated with MATLAB for the optimization of the real network topology of Evolink Bulgaria. It has been shown that increasing the number of SDN controllers has a significant effect on latency. When the number of controllers varies from one to three, a reduction of 2.0 ms to 1.5 ms is achieved for the latency in the worst case (worst case for three controllers).

Figure 4. Dependence of latency on the number of controllers.

2.6. Quality indicator (Figure of merit)
Network operators and ISPs are more concerned about saving capital costs - CapEx, related to networking. It therefore makes sense to take into account the cost of installing controllers for a given topology when determining their optimal number. In addition, in order to minimize the trade-off between productivity and costs, the latency of the distribution of packages under the worst conditions is also taken into account, according to Wald's well-known criterion.

This defines a qualitative indicator of the benefit of costs in a particular currency - \( CB \) by determining the ratio between a given number of controllers - \( k \), the cost of one controller per currency - \( C_k \) and the latency of packet propagation under the worst conditions - \( L_{\text{worst}} \), as shown in the equation below:

\[
CB = \frac{K \cdot C_k}{L_{\text{worst}}}
\]  

The cost-benefit indicator of a specific currency - \( CB \), related to the installation of new controllers in a given network is the comparison of the possible costs for controllers with the benefit (benefit, profit) that they contribute. This metric is crucial as it contributes to the overall CapEx and determines how much return on investment (ROI) is generated by network operators.

Figure 4 shows the simulation result with MATLAB for the worst case latency for 1, 2, 3 and 4 controllers.
From the simulation, the minimum value for latency in the worst case is. $L_{\text{worst}} = 1.5 \text{ ms} \ (0.0015 \text{ s})$ for 3 controllers.

In this case, the cost-benefit indicator in a specific currency - $CB$ is:

$$CB = 3 \cdot 1/1.5 = 2 \ [\text{costs in a specific currency/ ms}] \quad (6)$$

3. Results and discussion

MATLAB was used to simulate the built model for determining the optimal latency in software-defined networks. This model is designed to help developers who want to switch to SDN so that they can optimize network performance by calculating the optimal propagation latency and, accordingly, the number of controllers providing that latency when planning an SDN.

Network operators and ISPs are more concerned about saving capital costs - CapEx, related to networking. It therefore makes sense to take into account the cost of installing controllers for a given topology when determining their optimal number.

Algorithms are used to determine the optimal latency, which is the minimum latency for a specific number of controllers under the worst conditions (worst circumstances).

The developed algorithms are tested in the application and are visualized with the graphical capabilities of the software environment for numerical analysis MATLAB with a set of data on the network topology of the computer network of Evolink Bulgaria.

4. Conclusion

This article discusses the problem of determining the optimal latency in software-defined SDN networks with capital cost optimization - CapEx. A mathematical formulation for solving the problem is presented. The algorithms used to solve the problem are programmatically coded for simulation with MATLAB.

The simulation results show that the operation of only one controller leads to high propagation latency, as some network switches are located further away from the controller. A significant reduction in propagation latency was observed after simulation with two to three controllers.

However, the answer to the question of how many controllers to include in software-defined SDNs depends largely on the unique needs and constraints of each service provider. The article shows that the use of two controllers is the most effective way to achieve optimal latency in software-defined networks good QoS results at reasonable CapEx.

References

[1] Knight S, Nguyen H, Falkner N, Bowden R and Roughan M 2012 The Internet Topology Zoo Journal on Selected Areas in Communications 29 pp 1765-1775
[2] Kumari A, Member S and Sairam A 2019 A Survey of Controller Placement Problem in Software Defined Networks IEEE pp 11-13
[3] Lusani M, Joyce M and Albert L 2018 Optimum Placement of SDN Controllers in African Southern Africa Telecommunication Networks and Applications pp 1-7
[4] Zhang Q and Couloigner I 2005 A New and Efficient K-Medoid Algorithm for Spatial Clustering International conference on computational science and its applications pp 181-189
[5] Lusani M, 2018 Towards the Development of an Optimal SDN Controller Placement Framework to Expedite SDN Deployment in Emerging Markets Dept. of Electrical and Electronics Engineering University of Cape Town p 33
[6] Damak N 2010 Distance problems in networks theory and practice. Classical shortest-path algorithm pp 1-9
[7] Vasilev M, Delcheva S and Milusheva G 2019 Application of spectral characteristics for water stress analysis of cucumbers, Innovation and entrepreneurship 7 pp 1-7
[8] Kuchukova D 2017 Methods and algorithms for increasing the productivity of software-defined networks Dissertation “Angel Kanchev” University of Ruse pp 113-120