CHAPTER 9

Nature-Inspired Metaheuristic Approach to SD-WAN Controller Placement

Cindy Ujan and Mohd Murtadha Mohammad

9.1 INTRODUCTION

The Internet is the largest Wide-Area Network (WAN) that businesses use to connect their HQ or main offices with outlying branch locations. Employees and authorized users can connect using MPLS, broadband, or 4G LTE, as shown in Figure 9.1. However, the traditional WAN may hinder the businesses' ability to scale their network as their needs grow. The vertically integrated control and data planes in traditional network devices such as switches are mostly to blame for this (Kreutz et al., 2015). Additionally, implementing policy changes would require logging into each network device separately, which is time-consuming and laborious, especially in large-sized networks. Consequently, negatively impacts performance, cost, and complexity.

Software-Defined Networking (SDN) alters the deployment and management of these large-sized networks. SDN's control and data planes are separated and operate as distinct systems. Unlike traditional switches, SDN-compliant switches perform simple data-forwarding tasks that only need a small amount of processing power (Lu et al., 2019). This then makes them less expensive to purchase. With SDN's centralized control, businesses can anticipate cost savings, operational simplifications, and performance enhancements.

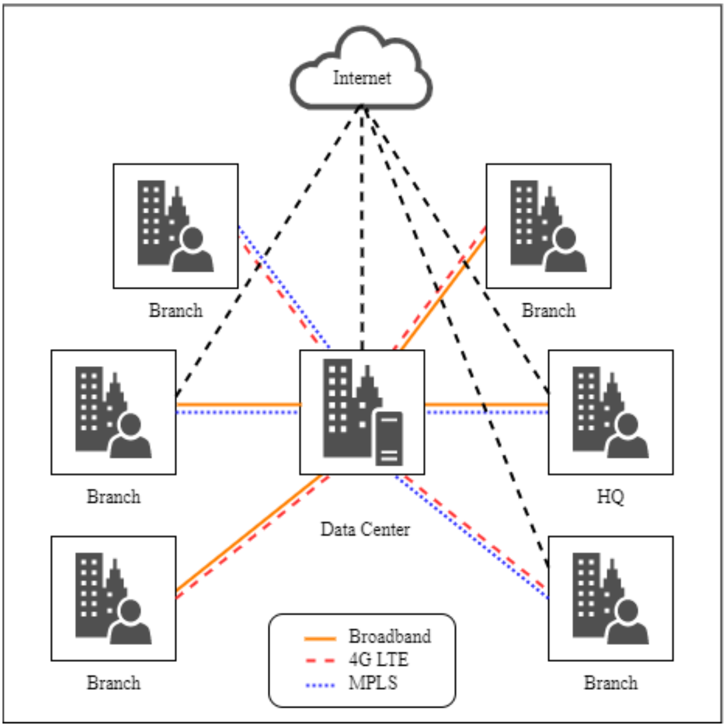

Figure 9.1 Example of an SD-WAN architecture between a business' HQ and branch offices

For the majority of medium-sized networks, a single SDN controller is adequate (Heller et al., 2012). However, a controller's capacity restricts the number of network requests it can handle. Thus, any rise in the number of network requests also increases the probability of a network bottleneck and Single Point of Failure. By introducing additional controllers, load balancing and backup redundancy solutions can be implemented to counteract such threats. How many controllers are required to support a large-sized SDN-based network, though? And where in the network should these controllers be placed? This puzzle is known as Controller Placement Problem (CPP). Additionally, academics continually look into and develop new, improved solutions because it is NP-hard. Also, because there are many different latencies and topologies in large-sized networks, determining where to put these controllers is essential (Wang et al., 2018).

Having been the first to introduce CPP, Heller et al. (2012) employed K-means clustering in their solution. K-means is easy to implement, quick to cluster, and simple to understand. However, K-means clustering has some known disadvantages, such as choosing the value of K, the random positioning of the initial centroids, and using Euclidean distance resulting in 2-D spherical solutions and one-point clusters due to outliers. Thus, K-means typically result in locally optimal solutions. Whereas metaheuristic algorithms can avoid the local optima and are adaptable (Mirjalili et al., 2017).

Based on recent and existing CPP solutions, this chapter introduces new objectives by proposing incorporating a metaheuristic solution to clustering while considering network (controller and link) failures. The chapter continues with Section 9.2, which presents the related works, and Section 9.3 summarizes the chapter.

9.2 RELATED WORKS

High-level methodologies such as metaheuristic algorithms can provide a thorough approach to dealing with optimization problems. Metaheuristic algorithms are, therefore, sophisticated search techniques (Harifi et al., 2020). These algorithms efficiently search the solution space by employing novel methodologies. For example, suppose the solution space is big, such as with large-sized networks. In those cases, the conventional approaches cannot find the solution in a reasonable length of time. The use of metaheuristic algorithms can solve this issue. Heuristic algorithms' main issues include premature convergence and getting stuck in local optima. Therefore, heuristic algorithm issues can also be resolved using metaheuristic algorithms.