Chapter 5
Cost–Effective Tabu Search Algorithm for Solving the Controller Placement Problem in SDN

Richard Isaac Abuabara
Universidad Santo Tomás, Colombia

Felipe Díaz-Sánchez
Universidad Santo Tomás, Colombia

Juliana Arevalo Herrera
Universidad Santo Tomás, Colombia

Isabel Amigo
IMT Atlantique, France

ABSTRACT

Software-defined networks (SDN) is an emerging paradigm that has been widely explored by the research community. At the same time, it has attracted a lot of attention from the industry. SDN breaks the integration between control and data plane and creates the concept of a network operating system (controller). The controller should be logically centralized, but it must comply with availability, reliability, and security requirements, which implies that it should be physically distributed in the network. In this context, two questions arise: How many controllers should be included? and Where should they be located? These questions comprise the controller placement problem (CPP). The scope of this study is to solve the CPP using the meta-heuristic Tabu search algorithm to optimize the cost of network operation, considering flow setup latency and inter-controller latency constraints. The network model presented considers both controllers and links as IT resources as a service, which allows focusing on operational cost.

INTRODUCTION

A software-defined network (SDN), is a network architecture that breaks the integration between the data and control planes of network equipment, as opposed to traditional networks that are vertically integrated. (Open Networking Foundation, 2016). The intention with this paradigm is to provide well-defined programming interfaces between the forwarding devices (switches) and a centralized entity (SDN controller) to enable software development to control the connectivity, network traffic and network function (Open Networking Foundation, 2013). This approach allows providing a unified "Network Operating System" (i.e. the controller) that allows simplifying the planning, deployment, commissioning, and management of the network.

An SDN network architecture consists of three layers (Figure 1):

- **The Application Layer**: groups network functions such as routing, load balancing, and switching.
- **The Control Layer**: where flow setup decisions are given as instructions to forwarding devices.
- **The Data Layer**: in charge of forwarding packets based on the instructions sent by The Control Layer. These instructions are saved in the Flow Table database.

Within the architecture, the control plane is composed of one or more controllers that act as a centralized entity to govern the operation of the entire network. This paradigm promises the possibility to have a high-level view of the network by providing a well-defined, and efficient communication protocol between the data layer and controller layer (Southbound interface). OpenFlow protocol (Mckeown et al., 2008) was created to that end, and it is the DE-facto standard for the Southbound interface.

Some advantages of SDN paradigm over traditional networks are:

- A centralized vision of the network, to deliver services and dynamism for physical and virtual device provisioning.
- Higher innovation since changes are made on software and not in hardware.
- Easier implementation of the complex function in the network using customized algorithms.
- Lower operative costs due to better server utilization, and better virtualization control.

The implementation of an SDN comes with different challenges, and one of them is the amount and physical location of the controllers., This problem is relevant to meet the needs of the network related to latency, performance, scalability, fault tolerance, security, among others, while maintaining a reasonable cost. To optimize the operation, it is necessary to establish an appropriate number of controllers, as well as the location for each controller within the network. In many cases, the parameters optimization will be a trade-off, so it is important to define thresholds as well as optimization objectives.

The calculation to obtain the number and location for the controllers is called the Controller Placement Problem (CPP) and it was first declared by Heller et al. (Heller, Sherwood, & Mckeown, 2012). The authors narrowed the problem to two questions: (1) How many controllers are needed? (2) Where in the topology should they go?.

Heller intends to address the CPP in a network that uses IT resources (controllers and links) as a service.

In this chapter, the authors propose an approach to find a solution to the CPP in SDN with the Tabu Search heuristic optimization method. The algorithm considers OPEX cost as the optimization objective, as well as flow setup and inter-controller latency, and feasibility constraints. First, a description of the

Related Content

iOS App and Architecture of Convolutional Neural Networks
(2020). *MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition: Emerging Research and Opportunities (pp. 1-22).*
www.irma-international.org/chapter/ios-app-and-architecture-of-convolutional-neural-networks/253272

Transformation Across Deep Learning Frameworks
(2020). *MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition: Emerging Research and Opportunities (pp. 62-109).*
www.irma-international.org/chapter/transformation-across-deep-learning-frameworks/253274

Mobile-Aided Breast Cancer Diagnosis by Deep Convolutional Neural Networks
(2020). *MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition: Emerging Research and Opportunities (pp. 145-162).*
www.irma-international.org/chapter/mobile-aided-breast-cancer-diagnosis-by-deep-convolutional-neural-networks/253277

Signal Processing and Pattern Recognition in Electronic Tongues: A Review
Jersson X. Leon-Medina, Maribel Anaya Vejarand Diego A. Tibaduiza (2020). *Pattern Recognition Applications in Engineering (pp. 84-108).*
www.irma-international.org/chapter/signal-processing-and-pattern-recognition-in-electronic-tongues/247793

Deep Learning Theory and Software
(2020). *MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition: Emerging Research and Opportunities (pp. 23-61).*
www.irma-international.org/chapter/deep-learning-theory-and-software/253273