An Intelligent Traffic Engineering Method over Software Defined Networks for Video Surveillance Systems Based on Artificial Bee Colony

Reza Mohammadi, Shiraz University of Technology, Shiraz, Iran
Reza Javidan, Shiraz University of Technology, Shiraz, Iran

ABSTRACT

In applications such as video surveillance systems, cameras transmit video data streams through network in which quality of received video should be assured. Traditional IP based networks cannot guarantee the required Quality of Service (QoS) for such applications. Nowadays, Software Defined Network (SDN) is a popular technology, which assists network management using computer programs. In this paper, a new SDN-based video surveillance system infrastructure is proposed to apply desire traffic engineering for practical video surveillance applications. To keep the quality of received videos adaptively, usually Constraint Shortest Path (CSP) problem is used which is a NP-complete problem. Hence, heuristic algorithms are suitable candidate for solving such problem. This paper models streaming video data on a surveillance system as a CSP problem, and proposes an artificial bee colony (ABC) algorithm to find optimal solution to manage the network adaptively and guarantee the required QoS. The simulation results show the effectiveness of the proposed method in terms of QoS metrics.

KEYWORDS

Video, Software defined network (SDN), Adaptive, Traffic Engineering, Artificial Bee Colony (ABC)

1. INTRODUCTION

Nowadays, due to growing multimedia data type, providing Quality of Service (QoS) for multimedia applications is a main agenda for network providers. Traditional networks are complex, and hard to manage dynamically (Kreutz et al., 2015), because network administrators must configure each individual network device separately. In traditional networks such as IP and ATM, network devices are vendor specific further control and data plane are coupled (Akyildiz, Lee, Wang, Luo, & Chou, 2014). Due to these limitations, there are a few successful traffic engineering methods for providing QoS in traditional networks. One of the most popular traffic engineering schemas in IP networks is MPLS (Xiao, Hannan, Bailey, & Ni, 2000). Nevertheless, MPLS is not suitable for real time applications such as online multimedia streaming (Egilmez, Civanlar, & Tekalp, 2013), because MPLS requires many signaling messages between routers for assigning labels and managing the routes themselves. IntServ (Clark, Shenker, & Zhang, 1992) and DiffServ (Ito & Tasaka, 2008) are other solutions for providing QoS in the traditional networks. These methods are based on hop by
hop manner of IP routing and do not care about overall network resources, hence they are not truly successful in all type of networks for multimedia applications (Egilmez, Civanlar, & Tekalp, 2013). Moreover, due to lack of global network view in legacy networks, it is difficult to guarantee QoS with respect to such constraints.

To overcome such limitations, Software Defined Networks (SDN) can be used. In SDN, the control plane and data plane of network are separated, and network providers will be able to control the whole network with a controller (Akyildiz, Lee, Wang, Luo, & Chou, 2014). Separation between control and data plane helps network administrators to manage the network dynamically. SDN provides network programmability, hence network administrators will be able to program and configure their networks per-flow and dynamically (Kim & Feamster, 2013). Figure 1 shows the simplified view of SDN architecture.

According to Figure 1, SDN comprises a number of data forwarding elements (switches) and a controller. The controller controls the forwarding elements by using southbound APIs. OpenFlow (McKeown et al., 2008) is one of the most popular southbound protocols widely used by SDN controllers. The controller can be a powerful single computer or server with software installed on it. Network administrator can program her/his desired algorithm or traffic engineering method in this controller software by using northbound APIs (Shin, M. K., Nam, K. H., & Kim, H. J., 2012). Then the controller controls the forwarding elements regarding the network administrator algorithms.

In SDN, an OpenFlow switch refers to a switch that supports OpenFlow protocol, and the controller can communicate and control the switch by using OpenFlow protocol. Communication between the switch and the controller can be optionally based on Transport Layer Security (TLS) (“The transport layer,”, 2016) protocol to achieve security. The OpenFlow switch has one or more flow table. Each flow table has some entries which contain a set of packet header fields, an action and one or more counters. The actions include drop, flood, send to the controller, send to specific port and etc. (Kreutz et al., 2015). Each entry of this flow table is filled by a forwarding rule which has received from the controller. When an OpenFlow switch receives a packet, it first checks the packet header field in forwarding rule table. If the packet header is matched a forwarding rule entry, the switch performs the action associated with that matched entry and updates its counter. Otherwise, the switch sends the header of the packet to the controller, and controller sends a new forwarding entry associated with that