Improving Resiliency in SDN using Routing Tree Algorithms

Kshira Sagar Sahoo, NIT, Department of Computer Science and Engineering, Rourkela, India
Bibhudatta Sahoo, NIT, Department of Computer Science and Engineering, Rourkela, India
Ratnakar Dash, NIT, Department of Computer Science and Engineering, Rourkela, India
Brojo Kishore Mishra, C. V. Raman College of Engineering, Department of Information Technology, Bhubaneswar, India

ABSTRACT

The ability to recover the control logic after a failure is detected in specific time window is called resiliency. The Software Defined Network (SDN) is an emerged and powerful architecture which allow to separate the control plane from forwarding. This decoupling architecture brings new difficulties to the network resiliency because link failure between switch and controller could defunct the forwarding plane. It has been identified that the resiliency of the network can be improved by choosing the correct place for the controller and by choosing proper routing tree once the controller location is known. In this work, we have analysed the performance of various Routing Tree algorithms on different network topology generated by Bernoulli Random Graph model and found that Greedy Routing Tree (GRT) provides the maximum resiliency. The Closeness Centrality Theorem has proposed to find the best controller position and later analysed the performance of various single controller placement algorithms on GRT for finding the overall improvement of the resiliency of the network.

KEYWORDS

Close-ness Centrality Theorem, GR, Software Defined Network, SPT

INTRODUCTION

It is tough to experiment on routers, on large scale productions as the equipment is too expensive and rate of innovation is very slow. The protocols defined on these routers lack in high levels of abstraction and standard interface. Since these are closed systems the innovation is only limited to the vendors who are actually building it which results in lack of new ideas in networking. This concept of traditional network cannot allow to dynamically changing the control function according to the situation. Millions of lines of code are baked into it and router only acts according to it. This results in many complex problems. To overcome these shortcomings of the traditional network, Software Defined Network is the correct solution for the future Internet technology.

SDN has many definitions. The Open Networking Foundation (ONF), is an organization which promotes OpenFlow interface of networking, defines SDN as the physical separation of control plane from the data plane (forwarding plane) in normal traditional networks (Kim & Feamster, 2013),
(Sahoo, Mohanty, Tiwary, Mishra, & Sahoo, 2016). ONF describes a high-level architecture of SDN, which functionally and vertically split into three layers (Specification, 2009).

- **Data Plane:** Data plane (Yassine, Rahimi, & Shirmohammadi, 2015) of the network typically consists of all the hardware equipment called network elements (NEs) like switches and the routers that are responsible for processing and delivering of packets between end points based on the states of the routers and endpoints.

- **Control Plane:** Control plane of the network is the intelligence behind NEs and is responsible to establish the states of the routers and endpoints and how packets should move from one NE to another to deliver to the endpoint (Sahoo, Sahoo, Dash et al., 2016).

- **Application Layer:** Application layer deals with end-user business applications that utilizes the SDN services. Business application such as energy efficient networking, security monitoring, network virtualization etc.

The network operating system or the controller, which is logically centralized, is the network intelligence behind the SDN and can be controlled using a programmable interface. A controller has the sole responsibility to manage the protocol, various network policies and installing flow rules on the network devices. The controller keeps up a worldwide perspective of the network system and this perspective is available through some well-defined open application programmable interfaces (APIs) to be utilized by various applications. There are many APIs like OpenFlow, ForCES, and OnePK available which can facilitate communication between the packet forwarding hardware and the network operating system or controller. But, OpenFlow is the foundational element for building various SDN solutions that was proposed to enable researchers to run experimental protocols in the campus networks (Jarraya, Madi, & Debbabi, 2014). Significant vendors have officially used this protocol in their monetarily accessible OpenFlow empowered switches. A case of an OpenFlow-based protocol characterizes messages, for example, send packet out, packet received, get stats, and modify forwarding table. This protocol establishes a secured channel between the controller and the switches during the communication. Multi-tenant data centre optimization, efficient network administration, programmability of the network are the major benefits of SDN. A simplified three layer SDN architecture is depicted in Figure 1. Note that, in the rest of the paper, we have used the term switch instead of an OpenFlow switch to ease the readability.

Figure 1. Three layers SDN architecture
JFeature: A Java Package for Extracting Global Sequence Features from Proteins for Functional Classification
www.igi-global.com/chapter/jfeature-java-package-extracting-global/76113?camid=4v1a

Using Biomedical Terminological Resources for Information Retrieval
www.igi-global.com/chapter/using-biomedical-terminological-resources-information/23055?camid=4v1a