Setting up an observer of a network of controller to detect anomalies

I. Context

Today's networks are managed by administrators based on metrological and monitoring information to react to changes in traffic and communication infrastructure failures. A possible architecture consists in automating this network management by deploying network controllers, dynamically capable of detecting and even anticipating the occurrence of changes in network system states and programming its reconfiguration as in the Software-Defined Networking (SDN) architecture. This centralized approach is increasingly used by companies such as Google, the world's number 1 network, which has part of its infrastructure under SDN.

SDN controllers are solely responsible for controlling the network. As a result, service on the network is degraded (even blocked) in the event of controller **failure or attack**. From a security point of view, it seems necessary to consider architectures with multiple controllers, thus avoiding the single point of failure/invasion and ensuring redundancy. Since even if a controller is only in charge of a subnetwork, it must have access to all network information to make decisions. Hence it is necessary to introduce a **communication interface between** controllers. This interface (named East-West) can be insecure, requires active maintenance of the connection, and **does not benefit from recognized standardized protocols**. Thus, an attack can be spread though this interface by the transmission of bad information in the same way as a BGP hijack.

II. Objective

To achieve this, we propose to integrate an **observer**, to **evaluate the likelihood of the decisions taken by the controllers**. This observer does not communicate with the controllers, which removes the threat of attack propagation via the interface between the controllers and the observer. The detection method will have to be adapted to cope with this new type of control, and one technique being considered is to define **a model of the behavior** of the consensus, to check that the behavior of each controller is likely regarding the consensus initially set up. The role played by the observer in this architecture is analogous to the role of **failure detectors**, as introduced in [Chandra96]. A possible avenue would be to formally define the requirements of such a failure detector and compare it to existing failure detectors in the literature.

However, the architecture will need to be tested to see whether one observer is sufficient, or whether a network of observers needs to be deployed. Due to the geographical layout and the quantity of data in particular, a **network of observers** might be considered, and there will be two types of difficulty: implementation (how to place the observers?) and the detection algorithm (how to ensure distributed detection?). The idea is to find a compromise between performance and cost.

Place of the internship: LIP, ENS de Lyon

Supervisors: Loïc Desgeorges; loic.desgeorges@univ-lyon1.fr

Skills required: network skills (TCP/IP stack, network performance, notion of QoS). Machine Learning knowledge is a plus. Fluency in English.

Starting date: between the beginning of February and the end of March 2024