Week 8 research note

From the thesis called ”Multiple controller management in software defined networking” I learned an algorithm used in this thesis that is a good way to achieve their scheme.

-Step1: Users submit the requirement of a network application. The application can be supported by deploying some virtual networks. The requirement for virtual networks is sent to the controller. The controller will choose appropriate network devices to deploy the virtual network.

-Step2: Network devices initiate connections to controller according to the number of connected controllers and their current local resource condition.

-Step3: The controller chooses to accept the connection initiated by network devices.

-Step4: After a connection is set up, controller sends message to get the network device’s information.

-Step5: If there are un-deployed virtual networks, then the controller will choose appropriate network devices and send deployment request messages to them. The deployment process will be carried out by using 2PC (two-phase commit).

-Step6: If the number of virtual networks that the network device doesn’t join in deploying is greater than a threshold, then the connection is tore down.

-Step7: When a connection is broken, after a random time interval, the network device will re-initiate a connection to the controller. Then go to step3.

By using the above algorithm, different controllers independently execute the above steps repeatedly. It resolves the problem that different controllers belonging to different vendors cannot be managed centrally.

Besides, if a network device often does not join in deploying of virtual networks, then it means that the network device is heavily loaded or there are not useful packet processing functions in this network device. In this case, the connection between controller and network device will be torn down. It is called an adaptive connection in their algorithm. As such, much overhead spent in computing and managing network device state can be saved.