Enhancement and Improvement

1. First improvement has been brought into the docker files of chat server and chat client (those are in ‘docker_file/chat_server’ and ‘docker_file/client_chat’ folders in).
   In the server side docker file, ftp server has been newly included.
   In the client side docker file, hping3 and ftp client has been newly installed.
   For creating image for server side, in ‘chat_server’ folder below command has been executed:

   ```
   “sudo docker build -t server .”
   ```

   Tag for docker image should be ‘server’. Otherwise Containernet topology has to be changed.
   For creating image for client side, in ‘client_chat’ folder below command has been executed:

   ```
   “sudo docker build -t client .”
   ```

   Tag for docker image should be ‘client’. Otherwise Containernet topology must be changed.

2. In slicing approach with RYU controller, before the improvement the slicing has been happened only on one application like for chatting only. But after the improvement slicing now can be done for both chatting application and also for file server application. For instance, now hosts with slice 1 only can get the service of chatting and hosts with slice 2 can only reach for the service of file server application. Moreover, experimental testing has been done for slicing on the basis on ICMP protocol and TCP protocol. For this scenario the updated RYU controller application can be found in ‘project_ryu’ folder named ‘slicing_ryu_controller.py’ file.

3. To enhance the scenario in number 2, an approach has been taken to detect and mitigate DDoS attack with ping flooding and TCP syn-ack flooding. Here, an open source IDS called snort has been used to detect the attack which then alert RYU controller. RYU controller then takes necessary steps to drop all the packets from that specific attacker. For executing this scenario slight change has been brought in the topology, where in access switch of data centre, a host has been included as a snort IDS. This topology file has been found named ‘topo_with_ryu_snort.py’ in ‘project_ryu’ folder. Moreover, RYU controller application for scenario in number 2 has also been taken care of, to provide the appropriate service. For designing RYU controller application for this scenario, all the traffic coming to data centre access switch have been mirrored to snort IDS port for monitoring. So, whenever any alert comes from snort IDS, RYU controller just drops all packet considering attacker’s source IP address with much higher priority. This RYU application file has been found named ‘slicing_application_ryu_controller_with_snort.py’ in ‘project_ryu’ folder. For the ruleset and thresholding snort IDS, files can be found in ‘project_ryu/ snort_rules_thresold’ folder with the name of ‘local.rules’ and ‘threshold.conf’ respectively. The rules for snort has been given below,

   ```
   alert icmp any any -> any any (msg:"icmp flood_block";detection_filter:track
   by_dst,count 5, seconds 1; sid:1000001;rev:1;)

   alert tcp any any -> any 21 (msg:"FTP syn";flags:S;detection_filter:track by_dst,count 5,
   seconds 1; sid:1000003;)

   alert tcp any any -> any 1060 (msg:"chat syn";flags:S;detection_filter:track by_dst,count
   5, seconds 1; sid:1000004;)
   ```