**Brief user manual describing how to run our code & tools:**

1. Launch Oracle VM Virtual box, followed by launching the Floodlight-Ubuntu VM, which is already configured with Mininet v.2.2.1, Floodlight v1.1, and OpenvSwitch
   * <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM>
2. Upgrade Floodlight v1.1 to Floodlight v1.2
   * <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343544/Installation+Guide>
3. Download and install D-ITG
   * <http://sdnopenflow.blogspot.com/2015/05/using-of-d-itg-traffic-generator-in.html>
4. Place our code folder (or git clone repo) inside the launched VM
   * Git clone repo from <https://github.com/TareqTayeh/Adaptive-Routing-Optimization-for-QoS-aware-Software-Defined-Networks-using-Deep-Reinforced-Learning>
5. To build and run the non-SDN network described in the paper:
   * Navigate to code by running “cd ~/Adaptive-Routing-Optimization-for-QoS-aware-Software-Defined-Networks-using-Deep-Reinforced-Learning/code/”
   * Run “sudo python advancedtopo\_no\_sdn.py” in terminal
   * Once the network is built, you will be prompted with the Mininet CLI. Run “sh ovs-ofctl add-flow <switch> action=normal” in the Mininet CLI for every single switch (s1, s2, … , s15) to manually add flows to the flow table and turn them into normal L2 devices
     + E.g. for switch 1: “sh ovs-ofctl add-flow s1 action=normal”
   * Now proceed to step 8
6. Launch the Floodlight controller in Terminal
   * (Only applies when you are running the SDN simulation)
   * <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM>
7. To build and run the SDN network described in the paper:
   * Navigate to code by running “cd ~/Adaptive-Routing-Optimization-for-QoS-aware-Software-Defined-Networks-using-Deep-Reinforced-Learning/code/”
   * Run “sudo python advancedtopo\_with\_sdn.py” in Terminal
   * You will then be prompted with the Mininet CLI, proceed with Step 8
8. To run the D-ITG flows for either network:
   * Using Mininet CLI, open xterm instances for each network host, including host 17 (ITG Log server)
   * Inside **each** xterm instance, go to where the D-ITG folder is installed
     + E.g. “cd ~/D-ITG-2.8.1-r1023/bin”
   * Initiate Log host on h17, the ITGLog Server
     + E.g. “./ITGLog”
   * Initiate each *even* host # from 2 to 16 as ITGRecv
     + E.g. “./ITGRecv”
   * Initiate hosts 1, 3, 5 and 11 as ITGSend
     + E.g. “./ITGSend <the\_associated\_quickflow\_script> -l <name\_of\_sender\_log\_file> -L 10.0.0.17 UDP -X 10.0.0.17 TCP -x <name\_of\_receiver\_log\_file>
   * All flows will be marked as finished when done
   * Terminate each ITGRecv host, followed by the ITGLog host (ITGSend hosts terminate by themselves after sending flows)
     + E.g.“^C”(to terminate each ITGRecv and ITGLog instances)
   * NOTE: If you have not moved the quickflow scripts from our code package into the D-ITG-2.8.2-r1023/bin folder, you will have to include the full path to their location (~/Adaptive-Routing-Optimization-for-QoS-aware-Software-Defined-Networks-using-Deep-Reinforced-Learning/code/D-ITG flow scripts/)
9. To decode and analyze the produced log files and generate a report
   * Can utilize ITGDec on the desired log file from any host
     + E.g. “./ITGDec <name\_of\_log\_file>”
   * To generate .dat files, which are utilized by ITGPlot
     + E.g. “./ITGDec <name\_of\_log\_file> <QoS\_metric> <time> <name\_of\_outputted\_.dat\_file>”
     + Where <QoS\_metric> is either -p (packet loss), -j (jitter), -d (delay), -b (throughput)
     + Where <time> is the sampling interval in milliseconds
10. To generate the plots via ITGPlot
    * Run “~/D-ITG-2.8.1-r1023/src/ITGPlot/ITGplot <input.dat> <number\_of\_the\_flow>”
      + Where <number\_of\_the\_flow> is an optional value, if nothing is indicated, all flows are plotted on the same graph
    * This generates a .eps file (the resulting plot file)
    * This file is located in “~/D-ITG-2.8.1-r1023/src/ITGPlot/”
11. Once all simulation activities are completed, exit the Mininet CLI “mininet> exit” and run “sudo mn -c” prior to running any further simulations to end and delete the simulation.

Additional Note: other D-ITG multi-flow scripts are also available in the code package (~/Adaptive-Routing-Optimization-for-QoS-aware-Software-Defined-Networks-using-Deep-Reinforced-Learning/code/D-ITG flow scripts/).