

Background

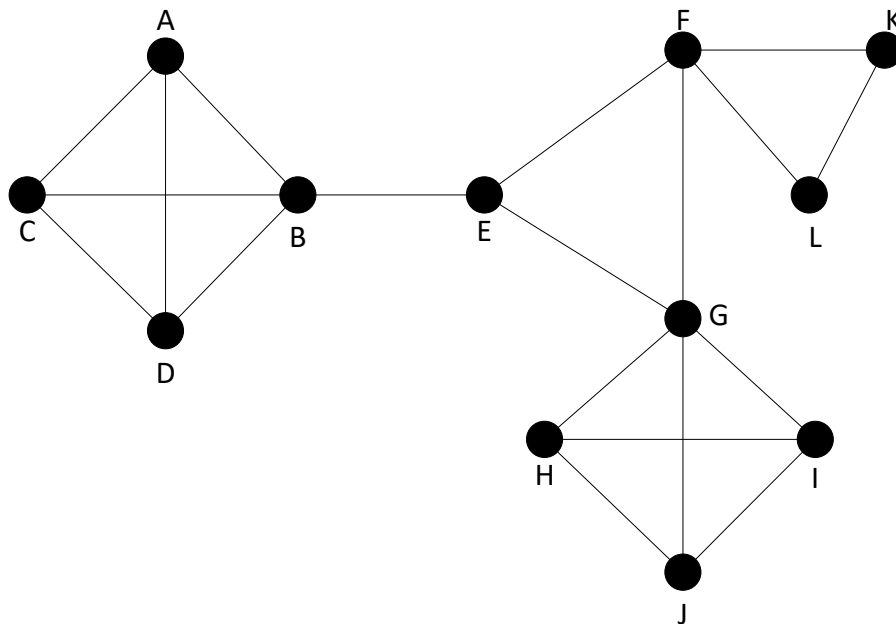
This project is an extension of the previous project and in this project you will perform a routing mechanism known as *flooding*.

Project Description

When a node receives a frame it compares its Fake IP Address with the destination IP address in the received frame. If **there is a match** it will print the frame. If **it is not a match**, then we have to execute the following flooding algorithm.

The Flooding Algorithm: Each node that receives a frame containing a destination IP address that does not match its Fake IP address will send the frame to all its neighbors other than from the one it received from.

The following topology (virtual) will be used to generate the necessary input files. The structure of the input files will be same as in Project 2.



Let us say that we have frames to send from A to J. A will send the frames to all its neighbors (nodes B, C, D). Each node (B, C, and D) will check its Fake IP address with the destination address in the received frame from A. It will not match since the destination address in the frame is the Fake IP address of node J. All nodes, (B, C, and D) will send the frame that they received from A to all its neighbors **other than** the node from it received the frame from (A in our example). For example, C

will send the frame to nodes D and B. Node D will send the frame to C and B, and B will send the frame to C, D, and E. Node E will send the frame to F and G and the process continues. When finally when node J receives the frame from G, J will just contents of the frame.

As you may observe there are many copies of the frame that keep circulating. As part of the project you need to design a protocol to **dampen flooding**. When developing protocols to dampen flooding you can modify the WLAN frames that are being sent. For example, you may choose to modify and use some of the fields that are not being used to keep track of frame ID and others.

This project requires to open 12 instances in different terminals to build the virtual topology. I highly suggest to use batch (windows) or bash (linux) script.

Constraints

1. The project is due on DATE by 11:59pm. Source files should be submitted to the digital drop box on <http://learn.ou.edu>.
2. This project will be done individually. Consultation in any form with other members of the class is strictly prohibited.
3. Your program should be documented thoroughly and **20%** of the project grade will be for documentation.
4. You are required to demonstrate the functionalities of your programs for grading. The schedule will be announced later. The demonstration will take about 10-15 minutes.