# Abstract : SDN controller

**Note** :
we have implemented more functionalities than suggested abstract. ( can be seen in **report** )

**Abstract :**

You have to design a software defined network controller.
After every 20 clocks you get an NxN adjacency matrix whose element A(i,j) if = 1, tells you whether an edge exists or otherwise. After every 200 clocks you get a traffic matrix of the order NxNxkxm where N is the number of nodes in the network, k is the number of service levels and m is the number of service instances at that service level for the particular source-destination pair under consideration.
When a traffic request arrives - you have to compute the shortest path in an application and pass on a string that emulates the nodes from the source to the destination.
You have to resolve deadlocks in a timely manner.
When you send the routing string the switch has only capacity for 10 such strings. Hence if a 11th request comes, then you have to delete the 1st request and replace it.Keep all the routes that you populate with the switch in the off FPGA memory only.
Design and implement a memory controller that is work-conserving.
The controller should be able to also create routes for protection that is edge disjoint from the work path. When a work path fails the destination node intimates the controller. The controller then populates the tables with the protection path.
The controller also informs switches of priority of handling services.
The controller also measures statistics such as average latency for each path, jitter and packet drops. It does so by sending a packet to the network with some timestamps (assume all switches can adhere to timestamps and have a central sync clock).