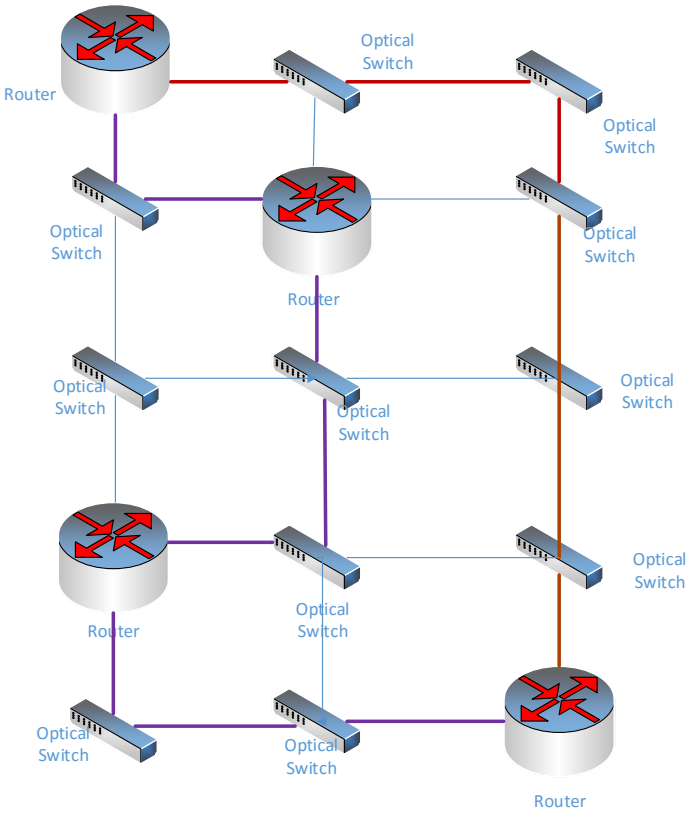
# Restorable Logical Topology using Cross-Layer Optimization


## Abstract:

Today's communication networks consist of routers and optical switches in a logical topology overlaid on an optical physical infrastructure. The routers are connected to each other via logical links called LightPaths, where each LightPath is established over one or more optical fibers and optical switches connecting these fibers.

The design of LightPaths involves selection of both logical link and logical path, i.e decision which pairs of routers will be connected by the LightPaths and how to route each LightPath across the optical network. Such design requires cross-layer optimization that will maximize the end-to-end traffic throughput in case of a physical failure in order to guarantee the restorability of the logical topology.

## Goals:

For a given network topology of routers and optical switches:

a. Build the topology using network simulator (ns-3 or Mininet).
b. Determine the LightPaths (Primary & Secondary) using Shortest Path Algorithm (Dijkstra's Algorithm**).**
c. Implement cross layer optimization algorithm that will:
    1. Decide which pairs of routers should be connected by a LightPath, while ensuring that the total number of LightPaths does not exceed a given budget of maximum number of LightPaths.
    2. Determine the physical path of each LightPath while ensuring that the number of logical links traversing a single physical link is minimum.
d. Compare between Dijkstra's Algorithm and the cross layer Algorithm in cases of link failure.

## Requirements:

C++ for ns-3

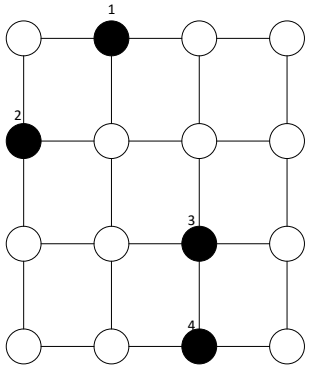
Python for Mininet

## Input:

- Physical Optical switches topology - represented as:
    - Gp=(Vp,Ep): Gp is Undirected Graph , Vp is set of the optical switches and Ep is set of optical links
    - Cp: Capacity of each link e ∈Ep. Cp>=2 and is even. Cp is same in all links.
    - Vl: Number of optical switches that can serve as routers -Subset of Vp
- Budget of the topology – represented as:
    - B : Maximum LightPaths allowed. Must be >= Vl

Input Example:

Gp:




Vp – 16 Optical Switches
Ep – 24 Optical Links
Vl – 4 Routers (1,2,3,4)
Cp(e) =2
B=4

Output:

a. LightPaths of given the given topology using Dijkstra's Algorithm
b. LightPaths of given the given topology using the Cross-Layer Optimization Algorithm – For example: