

# LAB 04B

## Xây dựng file NED cho mạng Fat-tree

### 1. Tạo ra một project Omnetpp trong đó file NED là một mạng fat-tree với $k = 4$ .

lab4b1.ned

```
simple Switch
{
    parameters:
        string layer;
        double pos_x = 0;
        double pos_y = 0;
        @display("p=$pos_x,$pos_y");
    gates:
        inout gate[];
}

simple Computer
{
    parameters:
        double pos_x = 0;
        double pos_y = 0;
        @display("p=$pos_x,$pos_y");
    gates:
        inout gate;
}

network Lab4b1
{
    @display("bgb=1000, 800");
    submodules:
        coreSwitch[4]: Switch {
            layer = "core";
            @display("i=block/switch");
        }
        aggSwitch[8]: Switch {
            layer = "aggregation";
            @display("i=block/switch,cyan");
        }
        edgeSwitch[8]: Switch {
            layer = "edge";
            @display("i=block/switch,yellow");
        }
        pc[16]: Computer {
            @display("i=device/pc");
        }
    connections:
        // Core layer <--> Aggregation Layer
        for i=0..3, for j=0..7 {
            coreSwitch[i].gate++ <--> aggSwitch[j].gate++ if (i < 2 && j % 2 == 0 || i >=
2 && j % 2 == 1);
        }

        // Aggregation layer <--> Edge layer
        for i=0..7, for j=0..7 {
            aggSwitch[i].gate++ <--> edgeSwitch[j].gate++ if (i >> 1==j >> 1);
        }
    }
}
```

```

    }

    // Edge layer
    for i=0..7 {
        edgeSwitch[i].gate++ <--> pc[i*2].gate;
        edgeSwitch[i].gate++ <--> pc[i*2+1].gate;
    }
}

```

lab4b1.cc: Chỉ dùng để set vị trí cho các nút

```

#include <omnetpp.h>
#include <stdio.h>
#include <string.h>
const double OFFSETX = 0;
const double OFFSETY = 50;
const double WIDTH = 1000;
const double HEIGHT = 800;

using namespace omnetpp;
using namespace std;

class Switch: public cSimpleModule {
protected:
    virtual void initialize();
};

Define_Module(Switch);

void Switch::initialize() {
    const char *layer = par("layer").stringValue();
    int i = getIndex();

    if (strcmp(layer, "core") == 0) {
        const double PADDING = 100;
        par("pos_x").setDoubleValue(
            OFFSETX + i * PADDING + WIDTH / 2 - (4 - 1) / 2.0 * PADDING);
        par("pos_y").setDoubleValue(OFFSETY);
    } else if (strcmp(layer, "aggregation") == 0) {
        const double PADDING = 100;
        par("pos_x").setDoubleValue(
            OFFSETX + i * PADDING + WIDTH / 2 - (8 - 1) / 2.0 * PADDING);
        par("pos_y").setDoubleValue(OFFSETY + 200);
    } else if (strcmp(layer, "edge") == 0) {
        const double PADDING = 100;
        par("pos_x").setDoubleValue(
            OFFSETX + i * PADDING + WIDTH / 2 - (8 - 1) / 2.0 * PADDING);
        par("pos_y").setDoubleValue(OFFSETY + 400);
    } else {
        EV << "Invalid layer name" << endl;
    }
}

class Computer: public cSimpleModule {
public:
    virtual void initialize();
};

Define_Module(Computer);

```

```

void Computer::initialize(){
    const double PADDING = 50;

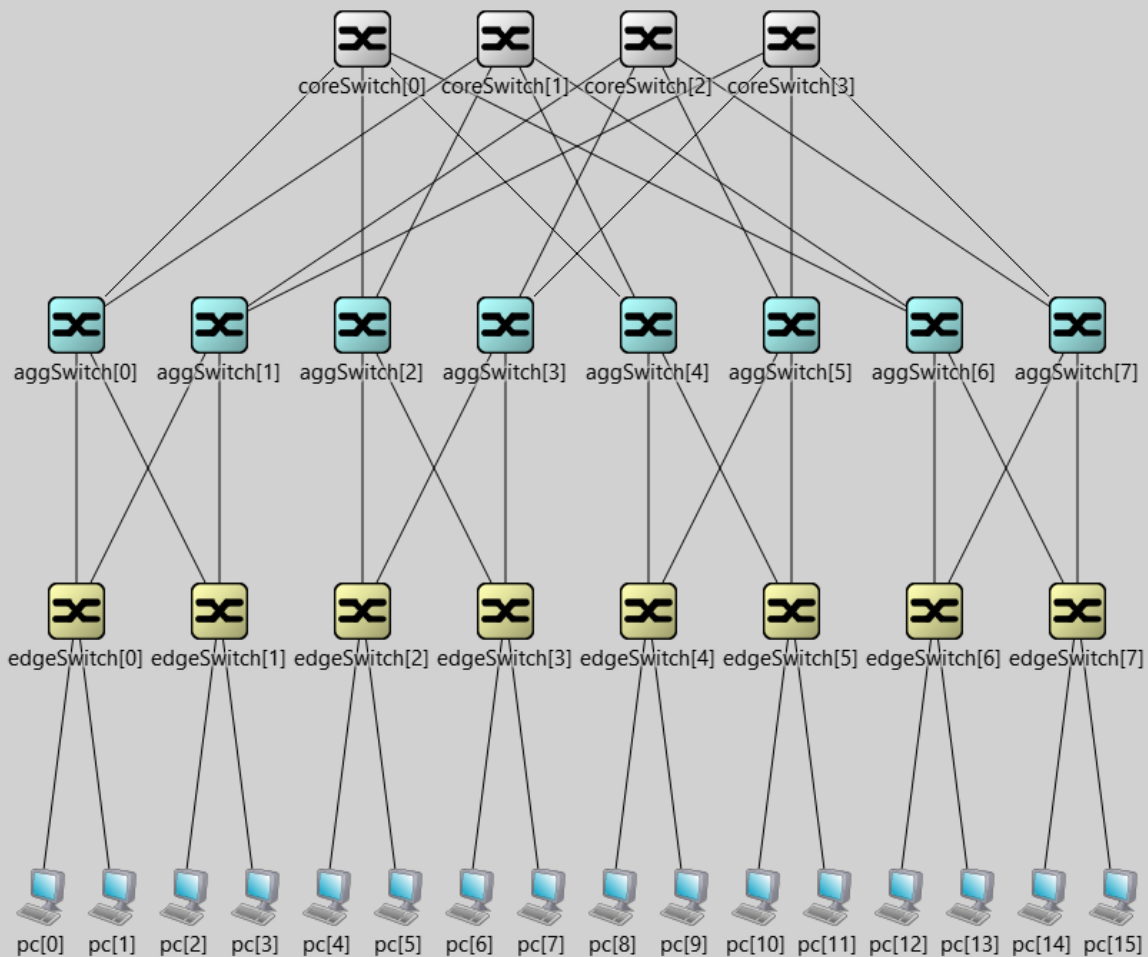
    int i = getIndex();

    par("pos_x").setDoubleValue(
        OFFSETX + i * PADDING + WIDTH / 2 - (16 - 1) / 2.0 * PADDING);
    par("pos_y").setDoubleValue(OFFSETY + 600);
}

```

kết quả

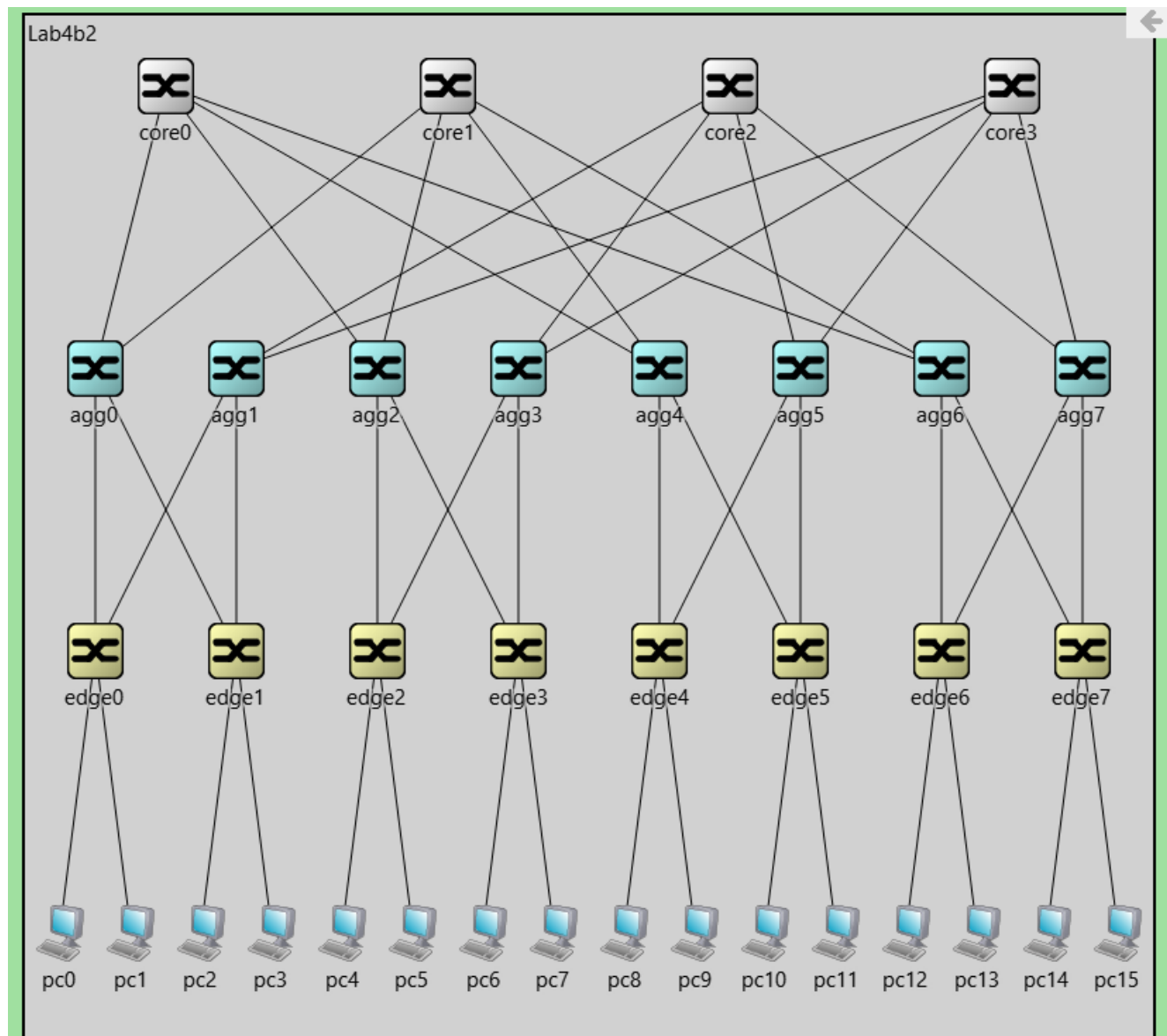
Lab4b1



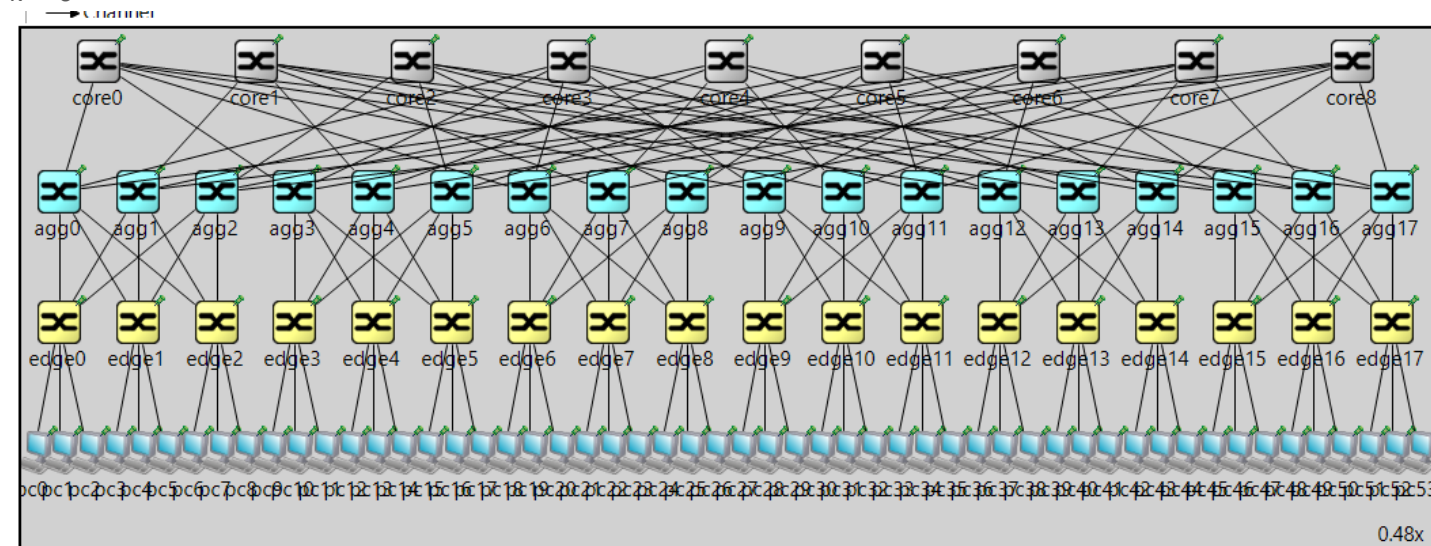
2. Xây dựng chương trình nhận đầu vào là giá trị  $k$ , và `dataRate`, đầu ra là mã nguồn file NED của mạng Fat-tree với giá trị  $k$  đó.

Source: <https://github.com/dranhclub/Omnetpp-project/tree/master/Lab%2004/Lab%2004B/Source>

k = 4



k = 6



k = 8

