**Lab 3: Đi sâu vào Tic-toc Tutorial (phần 2)**

Notes:

+ Có deadline

+ SV phải chụp lại các bức ảnh để minh chứng việc mình hoàn thành các bài tập trong từng Lab

+ Ghi câu trả lời (và cả ảnh chụp ra file pdf hoặc doc) rồi gửi đến địa chỉ email: **nguyenthanh@soict.hust.edu.vn** với tiêu đề mail và tên file như sau (chẳng hạn bạn sinh viên có tên là Nguyễn Văn An, mssv 20160001:

IT3920-STUDENT\_ID-NAME-Lab03 Vd.: IT3920-2016001-AnNV—Lab03

Chú ý, do ở Lab này có hai phần khác nhau với độ khó tăng dần, nên SV được phép làm lần lượt từng phần Lab03A và Lab03B và nộp theo các thời hạn khác nhau. Trong đó Lab03A hạn đến trước 12h trưa thứ ba tuần sau. Còn Lab03B có lẽ sẽ được nộp vào tuần sau nữa.

Ở phần này, SV tiếp tục hoàn thiện công việc của Part 4, 5, và 6 trong tic-toc tutorial và đi đúng theo các nhiệm vụ được yêu cầu trong các part đó. Tiếp theo trả lời các câu hỏi sau đây cũng như thực thi các công việc cần thiết (nếu có). Chụp lại các màn hình thao tác và các mã nguồn để minh chứng công việc của mình.

**Phần 1: Lab03A**

1. Giải thích ý nghĩa của các dòng mã nguồn dưới đây:

void [Txc10::forwardMessage](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/tictoc-tutorial/classTxc10.html#ac0a6e2b462a2615f6c0ce388ceec62a4)(cMessage \*msg)

{

// In this example, we just pick a random gate to send it on.

// We draw a random number between 0 and the size of gate `out[]'.

int n = gateSize("out");

int k = [intuniform](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/group__RandomNumbersDiscr.html#ga602a773dd9e4ba8b74f8aae19c36341c)(0, n-1);

[EV](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/group__Logging.html#ga650ef3eff8a2900bef69dae29c05d2dd) << "Forwarding message " << msg << " on port out[" << k << "]\n";

send(msg, "out", k);

}

1. Giải thích ý nghĩa của các dòng mã nguồn dưới đây:

if ([hasGUI](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/classomnetpp_1_1cComponent.html#af170a9e33793fd8ea42493f8f5b3fda3)()) {

char label[50];

// Write last hop count to string

sprintf(label, "last hopCount = %d", hopcount);

// Get pointer to figure

cCanvas \*canvas = [getParentModule](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/classomnetpp_1_1cModule.html#a6273eb514331e0e44130844d637b8b4c)()->[getCanvas](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/classomnetpp_1_1cModule.html#a3e2334a61989a6153b17e39f6fbbbcb3)();

cTextFigure \*textFigure =

[check\_and\_cast](http://127.0.0.1:56350/help/topic/org.omnetpp.doc/content/api/group__Utilities.html#ga5044f74639a1f5cd3701d438261a0407)<cTextFigure\*>(canvas->getFigure("lasthopcount"));

// Update figure text

textFigure->setText(label);

1. Đặt a và b là số ký tự trong tên và họ tên đệm của bạn, sau đó đặt X=min{a,b} và Y=max{a,b}.

a) Dựa trên ví dụ của *Tictoc10* hãy tạo ra một network có trên (X+Y) nút trong đó có X nút bậc 3 (nghĩa là có 3 láng giềng) và Y nút bậc 2 (2 neighbors). Chẳng hạn với lên Trần An thì network trong Tictoc10 có 6 nút, 2 có bậc ba và 4 có bậc 2.

b) Sửa mã nguồn sao cho mỗi gói tin (message) sẽ được gửi sao cho nó đi qua tất cả các hop (nút) và in ra giá trị hop count khi gói tin đến được nút cuối cùng. **Chú ý rằng người lập trình cho phép tùy chọn nút nguồn và nút đích qua file omnetpp.ini.**

1. Dựa trên câu hỏi 3, tạo ra mã nguồn sao cho mỗi nút nắm được các láng giềng trực tiếp kết nối đến nó. Kịch bản nắm bắt nút láng giềng như sau: tất cả các nút gửi gói tin đến với nút láng giềng của nó. Một khi một nút biết được tất cả các láng giềng của nó (3 hoặc 2 láng giềng tùy vào bậc) thì nút này in ra danh sách tên các láng giềng cũng như bậc của láng giềng.
2. Dựa trên câu hỏi 3, tạo ra mã nguồn sao cho sử dụng tin nhắn di chuyển qua tất cả các nút trong mạng, về lại nút ban đầu và in ra ma trận kết nối dạng 0 1 giữa các nút [i, j]. **Chú ý rằng người lập trình cho phép tùy chọn nút nguồn qua file omnetpp.ini.**

**Phần 2: Lab03B**

1. Kéo thả các nút vào đúng vị trí hợp lý sao cho mạng có 10 phần tử được hiển thị dễ nhìn nhất và xếp thành chữ cái đầu tiên của tên SV
2. Wireless Sensor Networks: Tạo ra 100 nút sensor trong không gian có kích thước 8\*8, các nút mạng có khoảng cách giữa hai nút bất kỳ <= 1. Viết mã nguồn tìm đường đi cho gói tin, một khi gói tin đến được một nút thì nó sẽ được gửi đến nút gần nhất (mà chưa đi qua) với nút hiện tại.

**Note:**

+ Có thể đọc các tham khảo hướng dẫn trong tài liệu Simulation Manual về việc gán tọa độ cho các nút.

*+ SV cần phải thêm nhiều nhất có thể các bức hình chụp trong quá trình làm việc với OmNET++*

**THAM KHẢO THÊM**

*Đoạn code sau thể hiện cách modify code để đặt tọa độ chi tiết cho từng node trong Tictoc10*

simple Txc10

{

    parameters:

        @display("i=block/routing");

        double xpos;

        double ypos;

        @display("p=$xpos,$ypos");

    gates:

        input in[];

        output out[];

}

network Tictoc10

{

    parameters:

        @display("bgb=951,473");

        tic[0].xpos = 300;

        tic[0].ypos = 100;

        tic[1].xpos = 300;

        tic[1].ypos = 200;

        tic[2].xpos = 300;

        tic[2].ypos = 300;

        tic[3].xpos = 600;

        tic[3].ypos = 100;

        tic[4].xpos = 600;

        tic[4].ypos = 200;

        tic[5].xpos = 600;

        tic[5].ypos = 300;

    submodules:

        tic[6]: Txc10{

        };

    connections:

        tic[0].out++ --> {  delay = 100ms; } --> tic[1].in++;

        tic[0].in++ <-- {  delay = 100ms; } <-- tic[1].out++;

        tic[1].out++ --> {  delay = 100ms; } --> tic[2].in++;

        tic[1].in++ <-- {  delay = 100ms; } <-- tic[2].out++;

        tic[1].out++ --> {  delay = 100ms; } --> tic[4].in++;

        tic[1].in++ <-- {  delay = 100ms; } <-- tic[4].out++;

        tic[3].out++ --> {  delay = 100ms; } --> tic[4].in++;

        tic[3].in++ <-- {  delay = 100ms; } <-- tic[4].out++;

        tic[4].out++ --> {  delay = 100ms; } --> tic[5].in++;

        tic[4].in++ <-- {  delay = 100ms; } <-- tic[5].out++;

}

Ngoài ra, để set tọa độ cho **tic[i]** thì trên modules ***Txc10*** ta còn có thể dùng **parameters**    như sau:

***double xpos;***

***double ypos;***

***@display("p=$xpos,$ypos");***