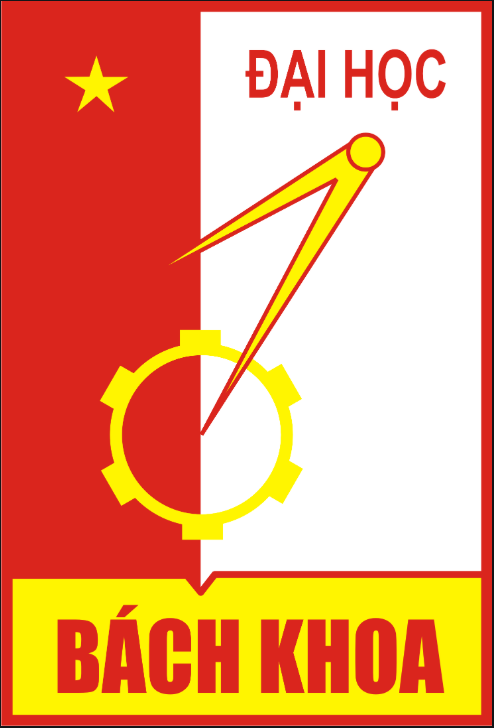


CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập – Tự Do – Hạnh Phúc

-----🙞🙜🕮🙞🙜----

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**BÁO CÁO MÔN HỌC PROJECT 2**

*Đề tài: Tính thông lượng của các loại hình data-center (torus2d và torus3d) sử dụng trình giả lập OMNeT++*

**Họ và tên: Nguyễn Huy Hoàng**

**MSSV: 20173132**

**Lớp: KHMT-05 K62**

**GVHD:**

*Hà Nội, 6 – 2020*

Lời cam kết

Tôi xin cam đoan đây là phần nghiên cứu và thể hiện đồ án môn học của riêng tôi, không sao chép các đồ án khác, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của khoa và nhà trường đề ra.

Lời cảm ơn

Em xin chân thành cảm ơn thầy giáo hướng dẫn Nguyễn Khanh Văn vì sự giúp đỡ và dìu dắt tận tình của thầy trong suốt quá trình em thực hiện đồ án.

Em xin cảm ơn thầy giáo hướng dẫn Nguyễn Tiến Thành vì những chỉ bảo của thầy đối với những vấn đề kỹ thuật mà em gặp phải.

Em cũng xin gửi lời cảm ơn đến các thầy, cô giáo trong viện CNTT&TT, trường đại học Bách Khoa Hà Nội, những nhà giáo đã truyền dạy cho em những kiến thức quý báu trong suốt những năm học vừa qua.

Cuối cùng, xin cảm ơn gia đình và bè bạn đã tạo điều kiện và luôn động viên tôi hoàn thành đồ án này.

Do thời gian hoàn thành đồ án có hạn cho nên những suy nghĩ cũng như sự thể hiện ý đồ không tránh khỏi có những khiếm khuyết. Em rất mong được sự động viên và đóng góp ý kiến của các thầy cô giáo.

Sinh viên thể hiện

Nguyễn Huy Hoàng

# Giới thiệu đề tài

## 1.1 Đặt vấn đề

Công nghệ thông tin ngày càng phát triển, nhu cầu của con người cũng ngày càng tăng lên. Một trong những vấn đề vướng mắc mà chúng ta đang gặp phải đó là vấn đề về mạng máy tính. Việc kết nối một vài nút mạng với nhau thì không là vấn đề, nhưng khi có hàng trăm, hàng nghìn hay thậm chí hàng triệu nút kết nối với nhau thì khi đó bài toán trở nên rất khó khăn. Sắp tới đây, thế giới sẽ chuyển sang kỷ nguyên của IoT – Internet of Things (Internet của vạn vật). Khi đó, vạn vật đều được kết nối với nhau thông qua mạng internet. Bài toán đặt ra là làm như thế nào để ta có thể kết nối chúng lại với nhau. Chia nhỏ ra thì ta có các bài toán con: Các thiết bị được kết nối theo hình trạng (topology) nào, sử dụng công nghệ kết nối nào, thuật toán định tuyến như thế nào, triển khai tầng vật lý ra sao, có đảm bảo được lưu lượng mạng hay không,... Và quan trọng là phải làm sao để tối ưu chi phí triển khai. Để giải quyết bài toán thì ta cần tính toán trên lý thuyết trước, rồi xây dựng mô hình để kiểm thử. Tuy nhiên, việc xây dựng mô hình thực tế (ví dụ như xây dựng một data-center) để kiểm thử là cực kì khó khăn, tốn kém, đặc biệt là với tình hình kinh tế của nước ta hiện nay thì nhiều trường hợp là bất khả thi. Chính vì thế, ta cần một bước trung gian giữa lý thuyết và thực tế, đó là xây dựng một mô hình giả lập, mô phỏng và tính toán trên đó. Công đoạn này sẽ giúp giảm thiểu những sai sót trong quá trình chuyển từ lý thuyết sang thực tế.

Để xây dựng một mô hình giả lập mạng, trong thực tế có nhiều công cụ có thể sử dụng. Tuy nhiên đồ án này sẽ chỉ sử dụng trình giả lập OMNeT++.

Bài toán cụ thể được lấy ra làm ví dụ ở đây là tính thông lượng (throughput) cho mạng data-center torus2d và torus3d.

## 1.2 Mục tiêu và phạm vi của đề tài

Đề tài này được thực hiện với một mục tiêu chính, đó là sinh viên có thể sử dụng thành thạo một trình giả lập. Ngoài ra còn một số mục tiêu phụ như là sử dụng thành thạo một / một số ngôn ngữ lập trình, biết cách phân tích yêu cầu, tự mày mò nghiên cứu với công nghệ mới để giải quyết yêu cầu được đặt ra; thêm hiểu biết về các khái niệm, giải thuật trong mạng máy tính, ...

Sản phẩm được yêu cầu sau khi hoàn thành đề tài đó là chương trình giả lập hoạt động của mạng data-center torus2d và torus3d. Cùng với đó là các phân tích số liệu về độ hiệu quả của các mạng data-center này.

## 1.3 Định hướng và giải pháp

Để thực hiện đề tài này, sinh viên đã được giáo viên hướng dẫn chia nhỏ công việc ra thành từng lab như sau:

1. Cài đặt OMNeT++ và nghiên cứu, thực hiện 3 mục trong Tictoc tutorial của trang chính thức của OMNeT++.
2. Đi sâu vào Tictoc tutorial
3. Thực hiện các mục còn lại trong Tictoc tutorial và xây dựng một mạng đơn giản có thể gửi gói tin qua lại giữa các nút.
4. Xây dựng file NED cho mạng fattree
5. Xây dựng bảng định tuyến cho Fat Tree và ngẫu nhiên cặp nguồn đích.
6. Mô phỏng quá trình truyền tin trong mạng trung tâm dữ liệu.

Các lab được đặt ra với độ khó tăng dần, lab cuối cùng là sử dụng tổng hợp kiến thức, mã nguồn của các lab trước đó để xây dựng lên chương trình hoàn chỉnh.

## 1.4 Bố cục báo cáo

Báo cáo được chia thành 6 phần chính như sau:

1. Giới thiệu đề tài
2. Khảo sát và phân tích yêu cầu
3. Công nghệ sử dụng
4. Triển khai và phát triển ứng dụng
5. Các giải pháp và đóng góp nổi bật
6. Kết luận và hướng phát triển

# Khảo sát và phân tích yêu cầu

## 2.1 Mô tả OMNeT++

### 2.1.1 OMNeT++ là gì?

OMNeT ++ là một thư viện và khung mô phỏng C ++ dựa trên thành phần, có thể mở rộng, chủ yếu để xây dựng các trình mô phỏng mạng. Mạng có nghĩa rộng hơn bao gồm các mạng truyền thông có dây và không dây, mạng trên chip, mạng xếp hàng, v.v. Chức năng dành riêng cho tên miền như hỗ trợ mạng cảm biến, mạng ad-hoc không dây, giao thức Internet, mô hình hiệu suất, mạng quang tử, v.v., được cung cấp bởi các khung mô hình, được phát triển như các dự án độc lập. OMNeT ++ cung cấp IDE dựa trên Eclipse, môi trường thời gian chạy đồ họa và một loạt các công cụ khác. Có các phần mở rộng cho mô phỏng thời gian thực, mô phỏng mạng, tích hợp cơ sở dữ liệu, tích hợp SystemC và một số chức năng khác. OMNeT ++ được phân phối theo Academic Public License

### 2.1.2 Các thành phần của OMNeT++

Các thành phần chính của OMNeT ++ là:

* Thư viện nhân mô phỏng (C ++)
* Ngôn ngữ mô tả cấu trúc liên kết NED
* IDE mô phỏng dựa trên nền tảng Eclipse
* GUI thời gian chạy mô phỏng tương tác (Qtenv)
* Giao diện dòng lệnh để thực hiện mô phỏng (Cmdenv)
* Các tiện ích (công cụ tạo makefile, v.v.)
* Tài liệu, mô phỏng mẫu, v.v.

### 2.1.3 Nền tảng

Hạt nhân mô phỏng OMNeT ++ là C ++ tiêu chuẩn và chạy cơ bản trên tất cả các nền tảng có trình biên dịch C ++ hiện đại. IDE mô phỏng yêu cầu Windows, Linux hoặc macOS.

### 2.1.4. Cách sử dụng

Dưới đây là tổng quan nhanh về cách ta sẽ sử dụng OMNeT ++:

1. Một mô hình OMNeT ++ được xây dựng từ các thành phần (mô-đun) giao tiếp bằng cách trao đổi thông điệp. Các mô-đun có thể được lồng nhau, nghĩa là, một số mô-đun có thể được nhóm lại với nhau để tạo thành một mô-đun ghép. Khi tạo mô hình, ta cần ánh xạ hệ thống của mình thành một hệ thống phân cấp các mô-đun giao tiếp.
2. Xác định cấu trúc mô hình trong ngôn ngữ NED. Ta có thể chỉnh sửa NED trong trình soạn thảo văn bản hoặc trong trình soạn thảo đồ họa của IDE mô phỏng OMNeT ++ dựa trên Eclipse.
3. Các thành phần hoạt động của mô hình (các mô-đun đơn giản) phải được lập trình trong C ++, sử dụng thư viện mô phỏng và thư viện lớp.
4. Cung cấp một omnetpp.ini phù hợp để giữ các tham số và cấu hình OMNeT ++ cho mô hình của ta. Một tập tin cấu hình có thể mô tả một số mô phỏng chạy với các tham số khác nhau.
5. Xây dựng chương trình mô phỏng và chạy nó. Ta sẽ liên kết mã với hạt nhân mô phỏng OMNeT ++ và một trong những giao diện người dùng OMNeT ++ cung cấp. Có dòng lệnh (lô) và giao diện người dùng đồ họa, tương tác.
6. Kết quả mô phỏng được ghi vào vector đầu ra và tập tin vô hướng đầu ra. Ta có thể sử dụng Công cụ phân tích trong IDE mô phỏng để trực quan hóa chúng. Các tệp kết quả dựa trên văn bản, vì vậy ta cũng có thể xử lý chúng bằng R, Matlab hoặc các công cụ khác.

## 2.2 Data center và topology

### 2.2.1 Data center

Data center (tạm dịch: trung tâm dữ liệu) là một công trình tập trung vào các hoạt động IT và các thiết bị của một tổ chức. Đây đồng thời cũng là nơi lưu trữ, quản lý và phân phối dữ liệu của tổ chức đó.

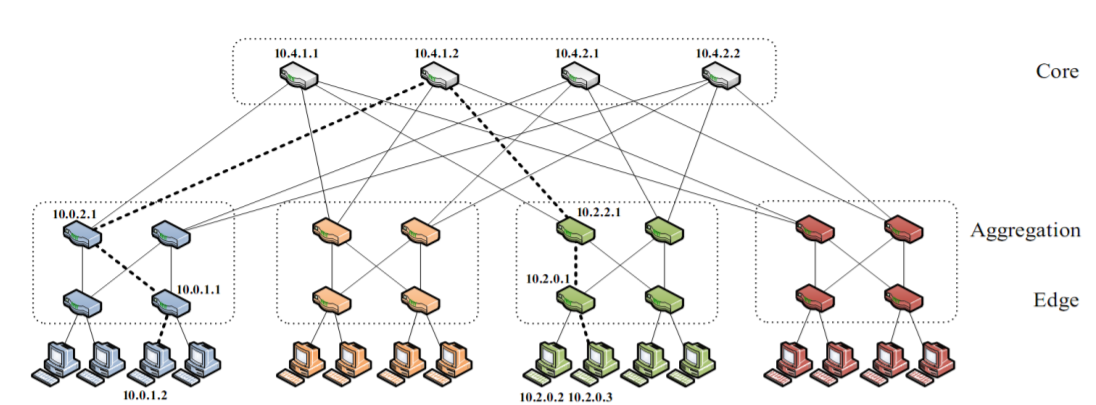
Hiểu đơn giản, data center là khu vực chuyên biệt chứa server hay phòng máy tính. Đây là nơi đặt, vận hành và quản lý server và và các thành phần liên quan (hệ thống truyền thông, hệ thống dữ liệu…).

Các server (và các thành phần liên quan) sẽ được kết nối với nhau thông qua một cấu trúc nhất định, gọi là topology.

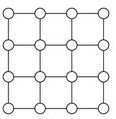
### 2.2.2 Topology

Topology của mạng là cấu trúc hình học không gian mà thực chất là cách bố trí phần tử của mạng cũng như cách nối giữa chúng với nhau. Ví dụ trong mạng LAN thường có 3 dạng cấu trúc là: Mạng dạng hình sao (Star Topology), mạng dạng vòng (Ring Topology) và mạng dạng tuyến (Linear Bus Topology).

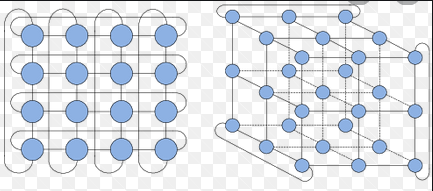
Dưới đây là một số hình ảnh trực quan về các topology hay được sử dụng trong các data center:



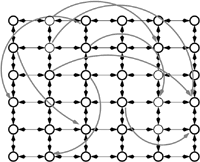
Hình 1: Fattree topology



Hình 2: Grid topology



Hình 3: Torus2d và torus3d

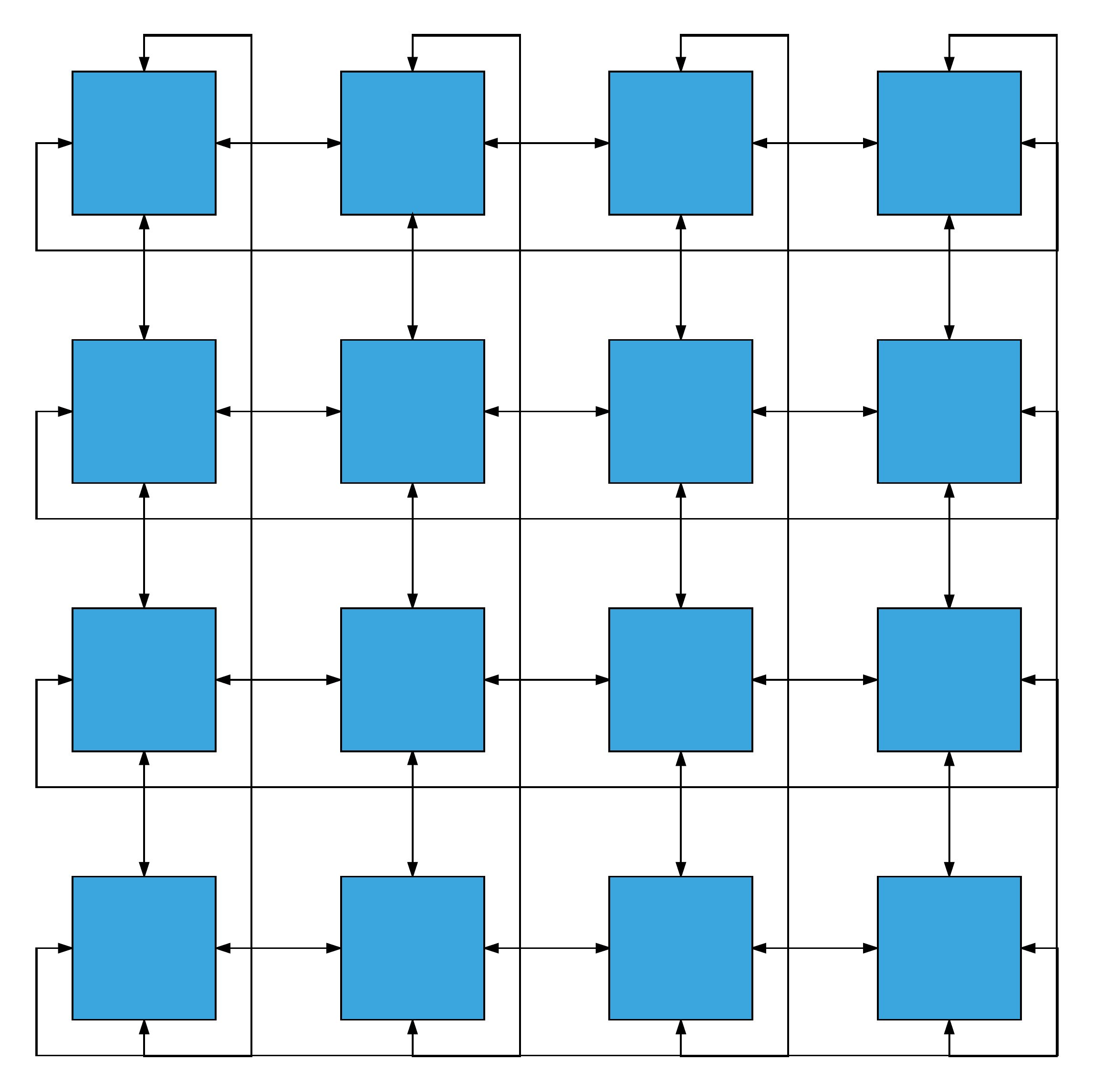


Hình 4: Random shortcut topology

Mỗi topology sẽ có những ưu điểm và nhược điểm khác nhau. Tuỳ thuộc vào các yếu tố mà người ta sẽ chọn một topology tương ứng cho data center, hoặc kết hợp chúng với nhau.

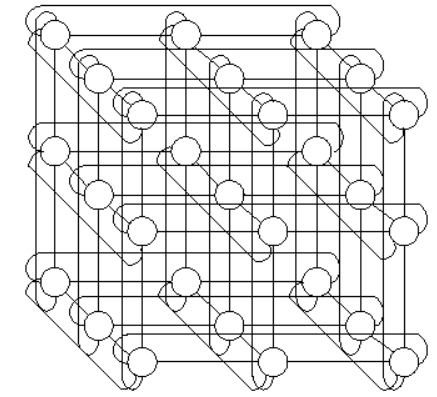
Trong phạm vi của đề tài, ta chỉ thực hiện mô phỏng và tính toán trên hai topology là torus2d và torus3d.

Trong mạng torus 2d (hình 5), các nút được bố trí thành hình vuông có cạnh N × N. Các nút cùng hàng, liền kề nhau sẽ được kết nối với nhau. Tại mỗi hàng, nút đầu tiên được kết nối với nút cuối cùng. Tương tự đối với các cột.



Hình 5: Torus 2D

Tương tự với torus 2d, torus 3d bao gồm các nút được bố trí thành khối lập phương có cạnh N×N×N. Các nút liền kề theo chiều dọc, ngang, sâu được kết nối với nhau, và các nút ở đầu mút có cùng chỉ số hàng, cột, sâu cũng được kết nối với nhau (hình 6).



Hình 6: Torus 3D

## 2.3 Giải thuật định tuyến

## 2.4 Phân tích yêu cầu

### 2.4.1 Xây dựng topo

### 2.4.2 Xây dựng giải thuật định tuyến

### 2.4.3 Công thức tính throughput

# Công nghệ sử dụng

Nói sơ về các NNLT và vai trò trong project

Java, NED, C++, Python

# Phát triển và triển khai ứng dụng

Mô tả biểu đồ lớp

# Các giải pháp và đóng góp nổi bật

## 5.1 Nghiệp vụ bài toán

## 5.2 Các khó khăn

# Kết luận và hướng phát triển

Tài liệu tham khảo

Trang chủ của OMNeT++

<https://omnetpp.org/>