**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO MÔN HỌC PROJECT 2**

Đề tài: Tính thông lượng của các loại hình data center (fat-tree) sử dụng trình giả lập OMNeT++

**MÔN HỌC: PROJECT 2**

**GVHD: PGS. TS. NGUYỄN KHANH VĂN**

**SINH VIÊN: NGUYỄN VĂN HỒNG**

**MSSV: 20173146**

**LỚP: CNTT-11 K62**

Hà Nội, tháng 6 năm 2020

**LỜI CAM KẾT**

Em xin được cam đoan đề tài: “Tính thông lượng của các loại hình data center (fat-tree) sử dụng trình giả lập OMNeT++” là phần nghiên cứu của riêng em dưới sự hướng dẫn của giáo viên hướng dẫn: PGS. TS. Nguyễn Khanh Văn và THS Nguyễn Tiến Thành hỗ trợ các vấn đề kỹ thuật trong trình giả lập OMNeT++. Nội dung báo cáo là sản phâm em nỗ lực nghiên cứu trong quá trình học môn học Project 2. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực. Em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu có bất cứ vấn đề nào xảy ra.

**LỜI CẢM ƠN**

Để hoàn thành để tài nghiên cứu: “Tính thông lượng của các loại hình data center (fat-tree) sử dụng trình giả lập OMNeT++” trước tiên em xin cảm ơn thầy Nguyễn Khanh Văn đã hướng dẫn tận tình giúp em hoàn thành công việc nghiên cứu, cùng với thầy Nguyễn Tiến Thành đã hỗ trợ kỹ thuật cho em rất nhiều trong việc sử dụng trình giả lập OMNeT++ để cho việc nghiên cứu được thực hiện suôn sẻ nhất có thể.

Em xin cảm ơn đến viện Công nghệ Thông tin và Truyền thông đã tạo điều kiện cho em được học và nghiên cứu đề tài này dưới sự hướng dẫn của thầy Nguyễn Khanh Văn và thầy Nguyễn Tiến Thành, cùng các thầy cô khác trong viện đã cung cấp cho em những kiến thức nền tảng để hoàn thành đề tài nghiên cứu này.

Do thời gian nghiên cứu làm đồ án chỉ trong một kỳ học, em cũng phải tiếp cận và nghiên cứu một mảng kiến thức mới nên không thể tránh khỏi những khuyết điểm, thiếu sót. Em rất mong sự góp ý của các thầy cô để em rút kinh nghiệm và hoàn thành môn học tốt hơn. Em xin chân thành cảm ơn!

Trân trọng, Hà Nội, Tháng 6 năm 2020

Sinh viên thực hiện

Nguyễn Văn Hồng

**MỤC LỤC**

**LỜI CAM KẾT…………………………………………………………...1**

**LỜI CẢM ƠN………………………………………………...…………...1**

**MỤC LỤC…………………………………………………………………2**

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

* 1. Đặt vấn đề
  2. Mục tiêu và phạm vi đề tài
  3. Định hướng giải pháp
  4. Bố cục báo cáo

**CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU**

2.1 Khảo sát hiện trạng

2.2 Giới thiệu về trình giả lập OMNeT++

2.2.1 OMNeT++ là gì?

2.2.2 Các thành phần của OMNeT++

2.2.3 Nền tảng

2.2.4 Cách cài đặt OMNeT++ trên Window

2.3 Phân tích yêu cầu kiểu hình mạng

2.3.1 Trung tâm dữ liệu (Data center)

2.3.2 Kiểu hình mạng (Topology)

2.3 Xây dựng mô hình mạng fat-tree

2.4 Xây dựng giải thuật định tuyến

2.5 Công thức tính thông lượng trung bình

**CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG**

3.1 C/C++

3.1.1 Giới thiệu về C/C++

3.1.2 Vai trò của C/C++ trong trình giả lập OMNeT++

3.2 NED (Network Description)

3.2.1 Giới thiệu về ngôn ngữ NED

3.2.2 Vai trò của NED trong trình giả lập OMNeT++

**CHƯƠNG 4: PHÁT TRIỂN VÀ TRIỂN KHAI ỨNG DỤNG**

**CHƯƠNG 5: CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT**

**CHƯƠNG 6: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN**

**TÀI LIỆU THAM KHẢO**

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

* 1. **Đặt ván đề**

Ngày nay sự phát triển của công nghệ thông tin là rất nhanh, cùng với đó là triển khai các dự án thực tế có quy mô lớn đòi hỏi phải tính toán tỉ mỉ, kỹ lưỡng trước khi đưa vào thực tế lắp đặt. Có rất nhiều dự án thất bại do chưa tính toán kỹ các rủi ro có thể gặp phải khi đưa ra mô hình mà không có sự kiểm thử thực tế kỹ lưỡng. Mạng máy tính là một trong những vấn đề đó khi chúng ta đưa ra mô hình mạng trên cơ sở lý thuyết không thì khó có thể tính toán được các thông số cần thiết của mạng đó. Nếu đưa mạng đó vào triển khai thực tế thì phát sinh chi phí rất lớn mà rủi ro cao, mô hình mà hoạt động không tốt đồng nghĩa với việc các phần cứng bỏ ra đều không còn giá trị sử dụng do đó rất tốn kém khi triển khai với mô hình thực tế.

Sự phát triển của Internet cũng như sự phát triển của kết nối mọi thứ thông qua Internet (Internet of Things - IoT - Internet kết nối vạn vật). Khi đó, mọi vật đều được kết nối với nhau thông qua mạng Internet. Vấn đề khi này là làm cách nào để kết nối các thiết bị với nhau, sử dụng mô hình nào để kết nối, thông tin được truyền đi như nào giữa các điểm hay là thiết bị, định tuyến cho gói tin sao cho đúng, đảm bảo hiệu quả của mô hình, triển khai thực tế thì cần những thiết bị gì kết nối với nhau bằng kết nối gì,… Có rất nhiều vấn đề nhỏ được đặt ra cần được giải quyết.

Cùng với việc triển khai thực tế gặp nhiều rủi ro về chi phí cũng như hiệu quả thực tế của mô hình. Do đó rất cần thiết một bước giả lập lại hệ thống sẽ hoạt động trong thực tế như nào để có được những số liệu cần thiết cho mô hình mạng khi muốn triển khai ra thực tế. Việc sai sót của mô hình là khó có thể tránh khỏi, công đoạn giả lập sẽ cho ra những kết quả tương tự như khi triển khai với các thiết bị thật từ đó phát hiện được những sai sót để khắc phục. Từ đó tối thiểu được các rủi ro có thể gặp phải khi triển khai thực tế.

Trong đồ án môn học này em sẽ sử dụng trình giả lập OMNeT++. Đây là một trình giả lập mạnh mẽ hiệu quả, dùng ngôn ngữ lập trình quen thuộc với nhiều người C/C++ để mô phỏng và tính toán.

Bài toán cụ thể là tính thông lượng cho mạng fat-tree.

* 1. **Mục tiêu và phạm vi đề tài**

Mục tiêu của đề tài là cung cấp cho sinh viên những kiến thức sau:

* Sử dụng thành thạo một trình giả lập (OMNeT++)
* Nắm vững kiến thức về ngôn ngữ lập trình C/C++ và một số ngôn ngữ lập trình khác
* Tìm hiểu được các mô hình hoạt động của các mạng trong trung tâm dữ liệu
* Học cách tự tìm hiểu, giải quyết các vấn đề gặp phải một cách nhanh và hiệu quả
* Khái quát được mô hình mạng và đặc điểm của từng mô hình đó
* Học được các giải thuật trong vấn đề định tuyến trong mạng máy tính
* Hiểu sâu hơn về hoạt động của mạng máy tính cũng như áp dụng vào các mô hình thực tế

Phạm vi của đề tài:

* Mô phỏng lại hoạt động của mạng fat-tree trong trung tâm dữ liệu
* Tính toán độ hiệu quả của mạng mang lại
  1. **Định hướng giải pháp**

Để hoàn thành đề tài này, sinh viên cần thực hiện những công việc nhỏ một như sau:

* Cài đặt OMNeT++ và nghiên cứu, thực hiện các hướng dẫn trong trang trính của OMNeT++ là các ví dụ làm mạng đơn giản Tictoc
* Đi sâu vào các bài hướng dẫn Tictoc
* Thực hiện phần còn lại của bài hướng dẫn Tictoc và xây dựng một mạng đơn giản có thể truyền tin giữa các nút
* Xây dựng file NED cho mạng fat-tree
* Xây dựng bảng định tuyến cho mạng fat-tree và sinh ngẫu nhiên cặp nguồn đích và đường đi định tuyến cho từng cặp
* Mô phỏng quá trình truyền tin trong mạng trung tâm dữ liệu

Các bài có độ khó tăng dần và bài cuối yêu cầu cần kiến thức và mã nguồn của các bài trước đó kết hợp lại để xây dựng nên một chương trình mô phỏng hoàn chỉnh.

* 1. **Bố cục báo cáo**

Bài báo cáo được chia làm các phần 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

**CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU**

**2.1 Khảo sát hiện trạng**

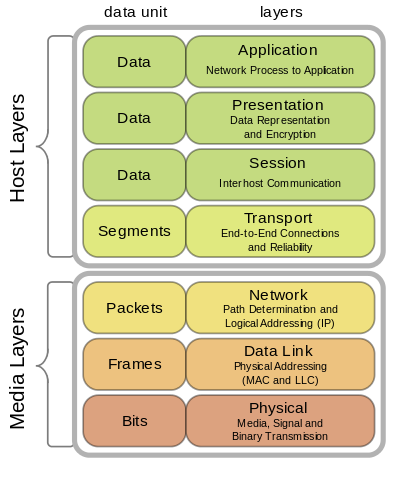
Chúng ta đang sống trong thời đại internet, các máy tính được kết nối với nhau tạo thành một mạng lưới. Chúng ta chỉ cần truy cập vào một địa chỉ nào đó, và như thế là 2 máy được kết nối với nhau. Tuy nhiên, đó là tầng ứng dụng, vậy tầng vật lý sẽ thực hiện như thế nào? Làm thế nào để có thể xác định chất lượng, sự khả thi nếu ta nghĩ ra một thuật toán định tuyến cho mạng? Trong báo cáo này, em sử cộng công cụ mô phỏng OMNeT++, một công cụ mô phỏng về định tuyến mạng mạnh mẽ mà em được nghiên cứu trong thời gian học môn Project 2.

Trước tiên em xin phép trình bày tổng quan về mô hình OSI.

Theo Wikipedia, thì mô hình OSI được định nghĩa:

Mô hình OSI (Open Systems Interconnection Reference Model, viết ngắn là OSI Model hoặc OSI Reference Model) - tạm dịch là Mô hình tham chiếu kết nối các hệ thống mở - là một thiết kế dựa vào nguyên lý tầng cấp, lý giải một cách trừu tượng kỹ thuật kết nối truyền thông giữa các máy vi tính và thiết kế giao thức mạng giữa chúng. Mô hình này được phát triển thành một phần trong kế hoạch Kết nối các hệ thống mở (Open Systems Interconnection) do ISO và IUT-T khởi xướng. Nó còn được gọi là Mô hình bảy tầng của OSI.

Mô hình OSI được chia thành 7 tầng như trong hình dưới đây:



Mô hình OSI

Tầng 1: Tầng vật lý (Physical layer) định nghĩa tất cả các đặc tả về điện và vật lý cho các thiết bị. Trong đó bao gồm bố trí của các chân cắm (pin), các hiệu điện thế, và các đặc tả về cáp nối (cable). Các thiết bị tầng vật lý bao gồm Hub, bộ lặp (repeater),thiết bị chuyển đổi tín hiệu(Converter), thiết bị tiếp hợp mạng (network adapter) và thiết bị tiếp hợp kênh máy chủ (Host Bus Adapter)- (HBA dùng trong mạng lưu trữ (Storage Area Network)).

Tầng 2: Tầng liên kết dữ liệu (Data-Link Layer) cung cấp các phương tiện có tính chức năng và quy trình để truyền dữ liệu giữa các thực thể mạng (truy cập đường truyền, đưa dữ liệu vào mạng), phát hiện và có thể sửa chữa các lỗi trong tầng vật lý nếu có. Cách đánh địa chỉ mang tính vật lý, nghĩa là địa chỉ (địa chỉ MAC) được mã hóa cứng vào trong các thẻ mạng (network card) khi chúng được sản xuất.

Tầng 3: Tầng mạng (Network Layer) cung cấp các chức năng và quy trình cho việc truyền các chuỗi dữ liệu có độ dài đa dạng, từ một nguồn tới một đích, thông qua một hoặc nhiều mạng, trong khi vẫn duy trì chất lượng dịch vụ (quality of service) mà tầng giao vận yêu cầu. Tầng mạng thực hiện chức năng định tuyến,.Các thiết bị định tuyến (router) hoạt động tại tầng này gửi dữ liệu ra khắp mạng mở rộng, làm cho liên mạng trở nên khả thi (còn có thiết bị chuyển mạch (switch) tầng 3, còn gọi là chuyển mạch IP).

Tầng 4: Tầng giao vận (Transport Layer) cung cấp dịch vụ chuyên dụng chuyển dữ liệu giữa các người dùng tại đầu cuối, nhờ đó các tầng trên không phải quan tâm đến việc cung cấp dịch vụ truyền dữ liệu đáng tin cậy và hiệu quả.

Tầng 5: Tầng phiên (Session layer) kiểm soát các (phiên) hội thoại giữa các máy tính. Tầng này thiết lập, quản lý và kết thúc các kết nối giữa trình ứng dụng địa phương và trình ứng dụng ở xa. Tầng này còn hỗ trợ hoạt động song công (duplex) hoặc bán song công (half-duplex) hoặc đơn công (Simplex) và thiết lập các quy trình đánh dấu điểm hoàn thành (checkpointing) - giúp việc phục hồi truyền thông nhanh hơn khi có lỗi xảy ra, vì điểm đã hoàn thành đã được đánh dấu - trì hoãn (adjournment), kết thúc (termination) và khởi động lại (restart).

Tầng 6: Tầng trình diễn (Presentation layer) hoạt động như tầng dữ liệu trên mạng. Tầng này trên máy tính truyền dữ liệu làm nhiệm vụ dịch dữ liệu được gửi từ tầng ứng dụng sang địng dạng chung.

Tầng 7: Tầng ứng dụng (Application layer) là tầng gần với người sử dụng nhất. Nó cung cấp phương tiện cho người dùng truy nhập các thông tin và dữ liệu trên mạng thông qua chương trình ứng dụng. Tầng này là giao diện chính để người dùng tương tác với chương trình ứng dụng, và qua đó với mạng.

Như đã nói ở trên thì OMNeT++ là một công cụ mô phỏng hết sức mạnh mẽ có thể mô phỏng lại toàn bộ bảy tầng của mô hình OSI. Do thời gian có hạn cũng như yêu cầu của đề tài nên bài báo cáo của em chỉ mô phỏng lại một số tầng cần thiết với yêu cầu.

**2.2 Giới thiệu về trình giả lập OMNeT++**

**2.2.1 OMNeT++ là gì?**

OMNET++ (Objective Modular Network Tested in C++) là một phần mềm mô phỏng nhằm thực hiện:

* Mô phỏng lưu lượng của một mạng viễn thông
* Mô phỏng các giao thức
* Mô phỏng mạng đa xử lý và phân bố rời rạc các hệ phần cứng
* Kiểm tra tính hợp lệ của cấu trúc phần cứng
* Đánh giá hoạt động của những hệ thống phần mềm phức tạp

Nó sẽ giúp mình mô phỏng 1 cái máy tính hoặc 1 con sensor. Để dễ hiểu hơn, chúng ta thử ví dụ. Bạn nghĩ ra một thuật toán định tuyến dùng trong mạng máy tính và bạn muốn kiểm chứng tính hiệu quả của thuật toán đó. Vậy chúng ta sẽ làm thế nào? Mình giả sử là thuật toán của bạn là thuật toán định tuyến. Vậy nếu bình thường bạn muốn kiểm chứng hiệu năng của thuật toán đó em cần phải làm cho card mạng của máy tính có thể chạy được thuật toán của bạn. Điều này gần như khôg thể làm được, vì thuật toán định tuyến của card mạng thường được cài đặt sẵn lúc sản xuất. Và OMNET++ đã được ra đời với các chức năng như trên để giải quyết vấn đề đã được trình bày. Nó sẽ giúp chúng ta kiểm chứng hoạt động các thuật toán mà mình đề xuất xem thuật toán của mình chạy có được không, so với các thuật toán khác thì hiệu năng có tốt hơn không.

**2.2.2 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.

**2.2.3 Nền tảng**

OMNeT++ sử dụng nền tảng C/C++ nên có thể chạy được ở hầu hết các môi trường được cài trình biên dịch này, Window, Linux, MacOs đều hỗ trợ.

**2.2.4 Cách cài đặt OMNeT++ trên Window**

Trước tiên ta cần tải bản OMNeT++ trên trang chính của OMNeT++ : “https://omnetpp.org/download/” . Sau khi tải xong và giải nén và thực hiện các bước như sau:

B1. OMNET++ được cài đặt bằng dòng lệnh, và để sử dụng, ta chỉ cần kích đúp vào file mingwenv.cmd

B2. Kiểm tra sự toàn vẹn của tệp tin configure.user bằng câu lệnh notepad configure.user. Nếu như nó tệp tin không toàn vẹn, ta chỉ cần download lại và làm từ đầu.

B3. Cài đặt các thư viện và các tiến trình xử lý của chương trình. Ta chỉ cần gõ 2 lệnh sau để build

./configure

make

B4. Mở OMNET++ IDE. Sau khi cài build các process và thư viện xong, ta chỉ cần gõ omnetpp để khởi động IDE, IDE của OMNET++ được xây dựng từ nhân của eclipse, do đó, các bước tương tự eclipse.

**2.3 Phân tích yêu cầu của kiểu hình mạng**

**2.3.1 Trung tâm dữ liệu (Data center)**

Trung tâm dữ liệu (Data center) là một hạ tầng tập trung nhiều thành phần tài nguyên mật độ cao (hardware, software…) làm chức năng lưu trữ, xử lý toàn bộ dữ liệu hệ thống với khả năng sẵn sàng và độ ổn định cao. Các hệ thống hạ tầng của trung tâm dữ liệu đều được dự phòng để đảm bảo sự hoạt động ổn định của hệ thống máy chủ bao gồm: Hệ thống mạng, hệ thống nguồn, hệ thống làm mát, hệ thống báo cháy báo khói, hệ thống quản lý vào ra, hệ thống Rack và CCTV…

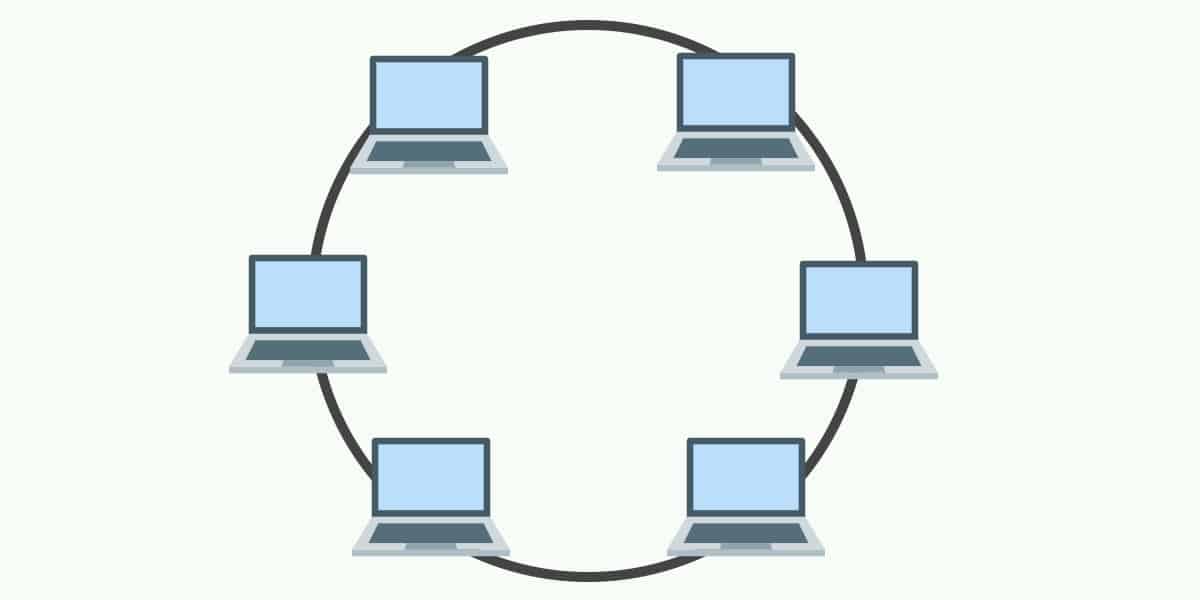
**2.3.2 Kiểu hình mạng (Topology)**

Topology của mạng là cấu trúc hình học không gian mà thực chất là cách bố trí phần tử của mạng cũng nh cách nối giữa chúng với nhau. Thông thờng mạng có 3 dạng cấu trúc là: Mạng dạng hình sao (Star Topology), mạng dạng vòng (Ring Topology) và mạng dạng tuyến tính (Linear Bus Topology). Ngoài 3 dạng cấu hình kể trên còn có một số dạng khác biến tướng từ 3 dạng này như mạng dạng cây, mạng dạng hình sao - vòng, mạng hỗn hợp,…

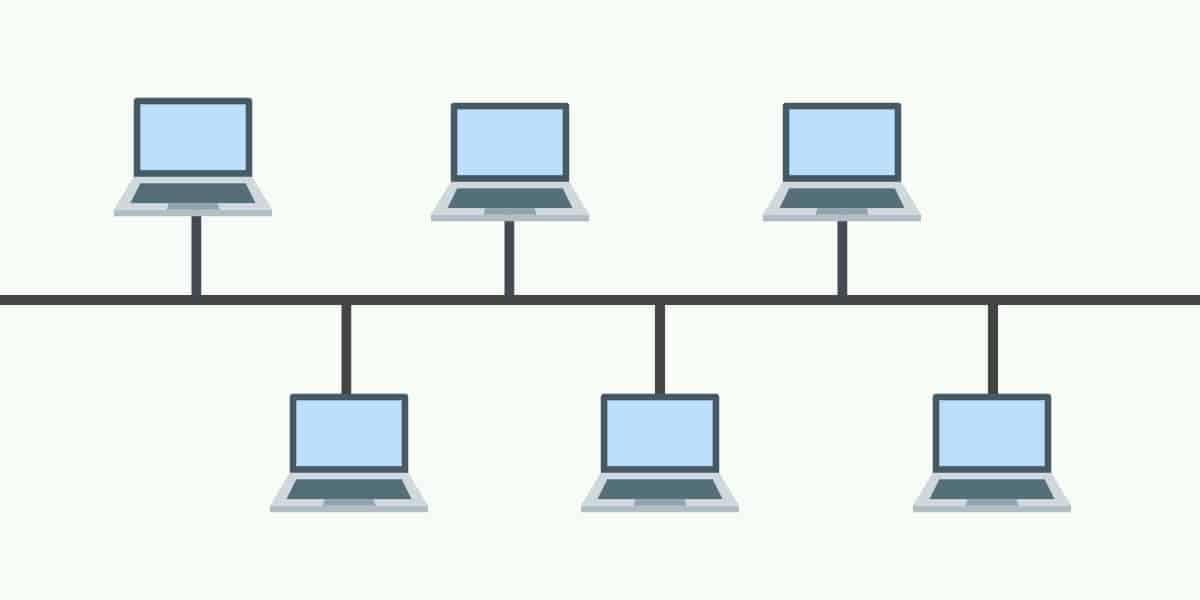
Một số kiểu hình mạng:



Mạng dạng hình sao (Star Topology)

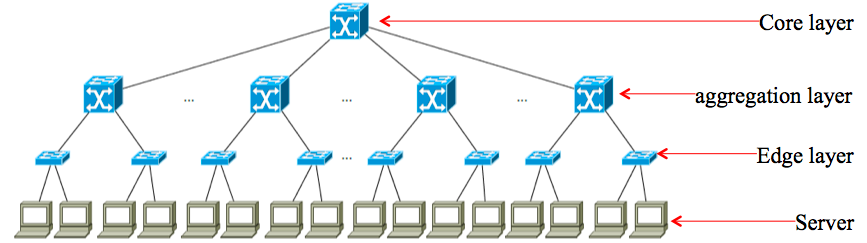


Mạng dạng hình vòng (Ring Topology)

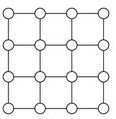


Mạng dạng tuyến tính (Linear Bus Topology)

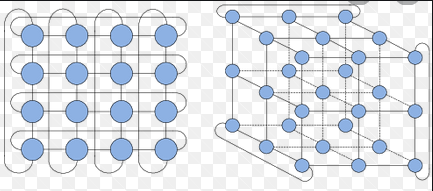
Một số mạng trong trung tâm dữ liệu:



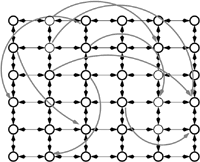
Mạng dạng fat-tree (Fat-tree Topology)



Mạng dạng lưới (Grid Topology)



Mạng torus 2D và 3D



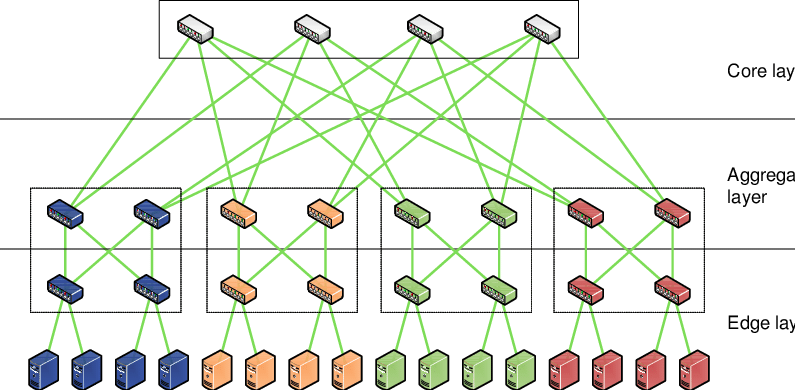
Mạng random shotcut

Có rất nhiều kiểu hình mạng (topology) khác nhau được sử dụng. Tùy vào mục đích sử dụng thì chọn một kiểu hình mạng hợp lý. Trong phạm vi môn học này em nghiên cứu về mạng fat-tree.

**2.4 Xây dựng mô hình mạng fat-tree**

Fat-tree là một cấu trúc mạng được sử dụng phổ biến trong các trung tâm dữ liệu, một mạng fat-tree được chia làm ba tầng khác nhau: Core, Aggregation, và tầng edge. Hình dưới mô tả kiến trúc của mạng fat-tree với số cổng của một switch là k = 4.

Ở kiến trúc Fat-tree, mỗi host (máy tính) kết nối với một switch. Mỗi switch ở tầng Core kết nối với k switch khác. Mỗi switch ở tầng Aggregation kết nối với k switch khác, còn mỗi switch ở tầng Edge thì kết nối với (k/2) máy tính và (k/2) switch khác. Số máy tính trong mạng Fat-tree là k\*k\*k/4 và số switch sẽ là: 5\*k\*k/4



**2.5 Xây dựng giải thuật định tuyến**

Bảng định tuyến là bảng ghi thông tin về đường đi đến một nút đích hoặc nhóm nút đích nào đó. Ở từng switch khác nhau sẽ có một bảng định tuyến. Với Fat Tree thì các switch cùng loại (cùng Edge Switch. Aggressive switch, và core switch) sẽ có bảng định tuyến giống nhau.

Từng switch sẽ xây dựng bảng định tuyến riêng cho mình khi khởi động. Khi có gói tin đến switch đó thì gói tin được bóc tách lấy thông tin của nút nguồn và nút đích, dựa trên bảng định tuyến xây dựng sẵn thì switch sẽ tìm ra nút kế tiếp trong đường đi đã được định tuyến. Do các bảng định tuyến ở các switch là giống nhau nên tại từng nút sẽ định tuyến ra cùng một đường đi từ nút nguồn tới nút đích.

**2.6 Công thức tính thông lượng trung bình**

Ta có công thức tính thông lượng trung bình (Throughput) của mạng theo từng khoảng thời gian (interval time) như sau:

Thông lượng trung bình cho biết được hiệu quả truyền tin trong mạng.

**CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG**

Trong đồ án này em sử dụng trình giả lập OMNeT++, ngôn ngữ chính chủ yếu sử dụng trong trình giả lập này là C/C++ và NED (Network Description).

**3.1 C/C++**

**3.1.1 Giới thiệu về C/C++**

C++ là một loại ngôn ngữ lập trình bậc trung (middle-level). Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại ưa thích và phổ biến của lập trình viên.

C++ được thiết kế hướng tới lập trình hệ thống máy tính và phần mềm nhúng trên các mạch vi xử lý, bao gồm cả hệ thống có tài nguyên hạn chế và tài nguyên khổng lồ, với ưu điểm vượt trội về hiệu suất, hiệu quả và tính linh hoạt cao. C ++ có thể tìm thấy ở mọi nơi, với những điểm mạnh là cơ sở hạ tầng phần mềm và các ứng dụng bị hạn chế tài nguyên. Bao gồm: phần mềm ứng dụng máy tính cá nhân, trò chơi điện tử, các hệ thống máy chủ (ví dụ: phần mềm thương mại điện tử, cỗ máy tìm kiếm trên web hoặc máy chủ SQL) và các ứng dụng ưu tiên về hiệu suất (ví dụ: tổng đài thông tin liên lạc hoặc thiết bị thăm dò không gian). C++ hầu hết được thực thi dưới dạng là một ngôn ngữ biên dịch, có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, Linux, Ubuntu và các phiên bản Unix. Nhiều nhà cung cấp cung cấp các trình biên dịch C ++, bao gồm Tổ chức Phần mềm Tự do, Microsoft, Intel và IBM.

**3.1.2 Vai trò của C/C++ trong trình giả lập OMNeT++**

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 là rộng hơn bao gồm các mạng truyền thông có dây và không dây, mạng trên chip, mạng xếp hàng,… Chức năng dành riêng cho tên miền như hỗ trợ mạng cảm biến, mạng ad-hoc, mạng không dây, giao thức Internet, mô hình hiệu suất, mạng quang tử,…, 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 Giấy phép Công cộng Học thuật (Academic Public License).

**3.2 NED (Network Description)**

**3.2.1 Giới thiệu về ngôn ngữ NED**

NED (Network Description) là ngôn ngữ mô tả cấu trúc liên kết của OMNeT++. Nó có một cú pháp đơn giản nhưng nó rất mạnh khi định nghĩa các cấu trúc liên kết thông thường như chuỗi, vòng, lưới, hypercube, cấu trúc cây,…

Các tính năng của ngôn ngữ NED

* Cấu trúc mô đun phân cấp, đánh máy
  + Viết loại mô-đun ghép một lần và tạo một số trường hợp
  + Độ sâu làm tổ không giới hạn
* Mô tả cấu trúc liên kết linh hoạt
  + Các tham số có thể định nghĩa: số lượng mô hình con, loại mô hình con, cấu trúc kết nối
  + Mẫu cấu trúc liên kết: một phương tiện tái sử dụng cấu trúc kết nối
* Thông số
  + Có thể chứa các biến ngẫu nhiên (do đó chúng có thể được sử dụng làm nguồn của các số ngẫu nhiên)
  + Có thể được thông qua bởi giá trị hoặc bằng cách tham khảo
  + Có thể chứa biểu thức của các tham số khác
* Hỗ trợ phân vùng linh hoạt của mô hình để thực hiện song song

**3.2.2 Vai trò của NED trong trình giả lập OMNeT++**

Network Description (NED) cho phép người dùng khai báo các module đơn, các kết nối và cách các module đơn ghép lại tạo thành module phức tạp.

Các khai báo ở NED giúp người dùng đặc tả được cấu trúc tĩnh của một mạng.

NED cho phép các module và channel có các interface cũng như các quan hệ kế thừa.