**Bài tập lớn: Mô phỏng quá trình truyền tin trong mạng trung tâm dữ liệu**

*Notes:*

*+ Có deadline*

*+ 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 email và tên file như sau (chẳng hạn bạn sinh viên có tên là Nguyễn Văn An, mssv: 20160001:*

***IT3920-STUDENT\_ID-NAME-BTL***

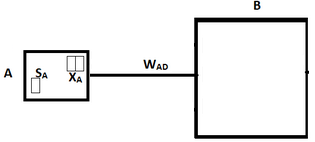
***Vd: IT3920-2016001-AnNV—BTL***

Ở bài tập lớn này, SV cần hoàn thành một chương trình Omnet++ cho phép giả lập được quá trình truyền tin trong mạng trung tâm dữ liệu. Chương trình này cài đặt một mô hình đơn giản trong đó có các nút nguồn (là nơi sinh ra gói tin để gửi), nút đích (là nơi nhận được các gói tin) và nút trung gian (nơi nhận rồi chuyển tiếp các gói tin). Khi gói tin đến được nút đích thì nút đích sẽ đếm gói tin này nhằm tính toán tổng dung lượng đã nhận được trong một khoảng thời gian. Khi kết thúc thời gian giả lập, tất cả dung lượng của các gói tin đến được các nút đích sẽ được dùng để tính toán được thông lượng của mạng theo thời gian. Chú ý trong việc truyền tin này ta mặc định là tất cả các gói tin đều được sinh ra theo một chu kỳ nhất định và một nút nguồn chỉ gửi đến một nút đích cũng như một nút đích chỉ nhận được gói tin từ duy nhất một nút nguồn. Bài tập lớn này có các phần việc được chia nhỏ ra như sau đây, nhằm giúp sinh viên hoàn thành được công việc tốt hơn:

**Các công việc dưới đây được chia nhỏ bài tập lớn. Các sinh viên cần tự đặt ra thời hạn để hoàn thành từng công việc sao cho chậm nhất đến tuần 16 là sinh viên có thể bảo vệ được kết quả của mình. Hàng tuần lên báo cáo với giảng viên để hai phía hiểu và đánh giá được tiến độ của sinh viên. Khi hoàn thành bài tập lớn thì SV cần nộp mã nguồn của từng công việc một. Chú ý rằng công việc số 5 cần sử dụng lại các mã nguồn của các công việc ở trước nó.**

**1. Cài đặt source queue và Exit buffer:**

Trong trường hợp tổng quát, một gói tin khi được sinh ra ở nút nguồn chưa chắc sẽ được gửi đi ngay mà còn căn cứ vào tình trạng của các thiết bị lân cận (chẳng hạn như dây dẫn có đang bận truyền gói tin khác không).



Hình 1: Source queue và exit buffer của nút nguồn

Công việc đầu tiên của bài tập lớn là cài đặt source queue (SQ) và exit buffer (EXB) cho một nút nguồn. Source queue là nơi lưu trữ các gói tin được sinh ra ở một nút còn EXB là nơi lưu trữ các gói tin chuẩn bị rời khỏi nút nguồn. SQ có kích thước vô hạn còn EXB là một mảng có kích thước giới hạn và kích thước này có thể là một tham số truyền vào cho file NED.

Công việc này yêu cầu sinh viên phải hoàn thành một project Omnet++ nhỏ, trong đó chỉ chứa hai nút nguồn đích A, B và đường truyền WAB như hình trên (trong hình vẽ giả sử chỉ có một phần tử trong SQ và EXB chỉ có kích thước là 2).

Các gói tin được sinh ra theo một chu kỳ nhất định, chu kỳ này cũng là tham số truyền vào file NED. Mỗi gói tin được truyền đi trên đường truyền sẽ mất một khoảng thời gian nhất định, giá trị khoảng thời gian này cũng là tham số được truyền vào file NED. Mỗi gói tin đều có một id của mình và id này là giá trị tự động tăng.

Mỗi nút đích đều có một mảng một chiều với phần tử thứ *i* lưu trữ số gói tin đến được nút đích tại interval time thứ *i*, giá trị interval time này cũng một tham số được truyền vào file NED

Một khi gói tin được sinh ra, nếu EXB còn chỗ trống thì gói tin được lưu trữ trong vị trí còn trống của EXB. Còn không thì gói tin vẫn nằm trong SQ.

Một khi EXB có chỗ trống mới thì một gói tin của SQ sẽ được chuyển lên cho EXB ngay.

Một khi vị trí đầu tiên của EXB có gói tin xuất hiện thì EXB sẽ kiểm tra xem đường truyền có bận không để truyền gói tin lên đường truyền.

Một khi đường truyền hết bận, nếu trong EXB có gói tin thì gói tin đó sẽ được gửi lên đường truyền.

Khi một gói tin đến được với nút đích, nút đích sẽ xác định thời điểm gói tin đến nơi, nếu thời điểm thuộc về interval time nào thì sẽ tăng giá trị của mảng một chiều tại phần tử interval time đó.

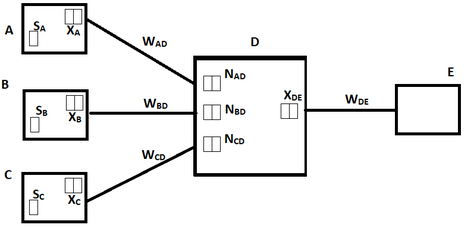
Khi kết thúc thời gian mô phỏng (thời gian này cũng là một tham số truyền vào file NED) thì chương trình in ra số lượng gói tin đến được nút đích trong từng interval time một.

Để dễ so sánh kết quả của SV, chương trình cần hoạt động với giá trị mặc định là chu kỳ sinh gói tin: 5ms. Thời gian truyền gói tin trên đường truyền cần mất 15ms. Giá trị interval time bằng đúng chu kỳ sinh gói tin. EXB có kích thước là 2. Thời gian giả lập là 1s

Gợi ý: SV có thể không cần thiết phải cài đặt source queue là một danh sách liên kết có độ dài vô hạn. Hãy nghĩ đến một cấu trúc dữ liệu đơn giản hơn khi đề bài đã cho các giải thuyết cực kỳ có lợi: (i) các gói tin chỉ được truyền đến một nút đích duy nhất. (ii) các id là tự động tăng, điều đó giúp cho việc xác định được số phần tử trong source queue chỉ bằng hai con số: id của gói cuối cùng và id của gói đứng đầu.

**2. Cài đặt Entrance buffer và Exit buffer ở các nút switch:**

Ở công việc tiếp theo, ta sẽ cài đặt một chương trình Omnet++ mô phỏng việc truyền tin trong một mạng đơn giản như sau: ba nút nguồn, một nút trung gian (switch) và một nút đích



Hình 2: ENB và EXB của nút trung gian

Như ở phần trước, bạn có thể thấy các gói t in trước khi rời khỏi nút nguồn cần phải đi qua một EXB. Tương tự, ở bài này gói tin cần phải đi vào một entrance buffer (ENB), sau đó được chuyển qua EXB rồi mới rời khỏi nút trung gian switch.

Kích thước của ENB và EXB của switch cũng được lấy từ tham số trong file NED. Để đơn giản, mọi nút switch đều có kích thước này giống nhau và cũng bằng kích thước của EXB trong nút nguồn. Nếu nút switch có kết nối đến với k nút khác nhau thì trong switch đó có tới k ENB và k EXB để phục vụ việc nhận/gửi gói tin qua k cổng khác nhau. Trong hình 2, để đơn giản thì hình vẽ này đã được lược bỏ bớt đi (không vẽ vào) các ENB và EXB còn lại.

Cơ chế hoạt động của mạng trong hình 2 như sau:

Điều kiện một gói tin được phép rời khỏi EXB của nút nguồn sẽ bổ sung thêm ý sau: nếu ENB của nút switch D còn chỗ trống thì gói tin mới được đưa lên đường truyền.

Một khi có một gói tin được gửi lên đường truyền, nút nguồn sẽ lưu trữ thông tin về số lượng các chỗ trống còn lại trong ENB của switch, (giả sử đây là biến mang tên counter). Rõ ràng khi truyền đi một gói tin thì biến counter này sẽ giảm đi một.

Một khi gói tin đến được vị trí đầu tiên của ENB trong switch, switch sẽ xem gói tin này cần đi ra cổng EXB nào của nó; có bao nhiêu gói tin cũng muốn đi ra cổng đó (nếu nhiều hơn 1 thì chọn gói tin nào có id bé nhất); cổng EXB kế tiếp đã đầy hay chưa?

Một khi các điều kiện trên thỏa mãn thì switch sẽ tạo ra một sự kiện với thời điểm dự kiến thực thi sự kiện là bằng thời điểm hiện tại cộng với chu kỳ hoạt động của switch (chu kỳ này cũng là tham số từ bàn phím). Sự kiện này sẽ truyền gói tin đó từ ENB sang EXB.

Một khi gói tin được truyền từ ENB sang EXB, một sự kiện sẽ xảy ra ở nút trước đó (A hoặc B hoặc C tùy vào gói tin xuất phát từ nút nào), biến counter sẽ được tăng thêm một. Thời điểm dự kiến xảy ra sự kiện này là một giá trị CREDIT\_DELAY được khai báo là tham số trong file NED.

Một khi gói tin đến được vị trí đầu tiên của EXB của switch, switch sẽ xem đường truyền có trống không? Nếu có thì gói tin của EXB sẽ được truyền sau một khoảng thời gian là một chu kỳ hoạt động của switch.

Một khi EXB chuyển từ trạng thái đang đầy sang trạng thái có một chỗ trống mới thì switch sẽ kiểm tra trong danh sách các gói tin ở phần đầu của các ENB, xem có gói tin nào muốn đến EXB hay không? Nếu có thì chọn ra gói tin có id nhỏ nhất mà kích hoạt sự kiện chuyển gói tin đó sang EXB. Tất nhiên, thời điểm dự kiến xảy ra sự kiện là thời điểm hiện tại cộng với một chu kỳ hoạt động của switch.

Chú ý rằng những sự kiện chuyển từ ENB sang EXB vẫn có thể không được thực hiện mặc dù đã đến đúng thời điểm dự kiến xảy ra của nó. Nguyên nhân hàng đầu là vì một lý do nào đó khiến EXB bị đầy, khi ấy sự kiện đó không làm thay đổi vị trí của gói tin (vẫn lưu trữ tại ENB).

Khi kết thúc thời gian mô phỏng (thời gian này cũng là một tham số truyền vào file NED) thì chương trình in ra số lượng gói tin đến được nút đích trong từng interval time một.

Để dễ so sánh kết quả của SV, chương trình cần hoạt động với giá trị mặc định là chu kỳ sinh gói tin: 5ms, đó cũng là chu kỳ hoạt động của switch. Thời gian truyền gói tin trên đường truyền cần mất 15ms. Giá trị interval time bằng đúng chu kỳ sinh gói tin. Giá trị Credit\_delay là 1ms. EXB có kích thước là 2. Thời gian giả lập là 1s

**3) Sinh tự động mạng trung tâm dữ liệu:**

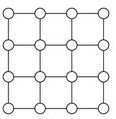
Ở trung tâm dữ liệu, các nút nguồn đều có các source queue, exit buffer, counter. Các nút trung gian đều có các ENB và EXB ở bên trong. Bản thân nút trung gian nếu kết nối đến với m switch khác thì sẽ có tổng cộng m biến counter dành riêng cho từng switch. (SV có thể cho các biến này là thuộc tính trong mã nguồn NED hoặc mã nguồn C++ - tùy).

Mỗi sinh viên sẽ cần phải cài đặt một topology mạng khác nhau. Các topology này được lưu trữ trong file NED và file NED này được sinh tự động qua một mã nguồn ngôn ngữ tùy chọn của sinh viên:

Với các bạn Nguyễn Văn Hồng và Đỗ Đăng Việt Hoài sẽ xây dựng chương trình sinh ra file NED tự động cho mạng Fat-tree với k là giá trị đầu vào.

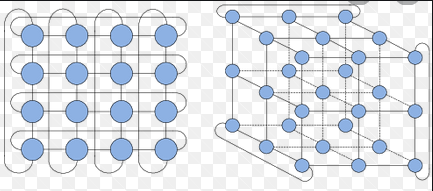
Ngoại trừ mạng Fat-Tree, các mạng dưới đây đều có bố trí mọi nút switch đều có kết nối với một host khác (mà các host không được vẽ trong các hình dưới)

Bạn Nguyễn Quang Linh sẽ xây dựng chương trình sinh ra file NED tự động cho mạng grid-2D với N là số switch tại mỗi chiều (ngang dọc) (hình 3).



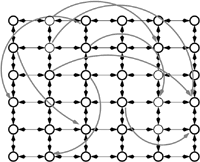
Hình 3: mạng Grid-2D với số N = 4

Bạn Nguyễn Tuấn Nam sẽ xây dựng chương trình sinh ra file NED tự động cho mạng Torus-2D với N là số switch tại mỗi chiều ngang dọc. Bạn Nguyễn Huy Hoàng, Nguyễn Bá Quân sẽ xây dựng với Torus-3D, trong đó cần N là số switch cũng tại mỗi chiều ngang dọc.



Hình 4: Mạng Torus-2D và torus-3D.

Bạn Nguyễn Văn Ngà sẽ xây dựng chương trình sinh ra file NED tự động cho mạng random shortcut network. Đây là một mạng có dạng giống Grid, nhưng bên cạnh giá trị N đặc trưng cho số switch tại mỗi chiều ngang dọc thì mạng này lại có giá trị k là số lượng các đường kết nối ngẫu nhiên từ một switch bất kỳ đến k nút khác. Hình 4 minh họa một mạng random shortcut network như vậy với N = 6 và k = 1. Chú ý rằng một số switch trong hình này đã không vẽ đường kết nối ngẫu nhiên vì lý do tránh việc có quá nhiều đường cong trong hình (chứ trên thực tế vẫn có các đường kết nối).



Hình 5: mạng random shortcut network

**4) Viết mã nguồn của cho chức năng định tuyến:**

Ứng với mỗi topology dành riêng cho từng SV, mỗi SV cần xây dựng bảng định tuyến và hàm phục vụ chức năng định tuyến cho gói tin. Một khi gói tin đi đến một nút trung gian, switch đó sẽ kiểm tra xem gói tin này cần đi đến nút đích D nào. Sau đó kiểm tra trong bảng định tuyến của mình để biết xem gói tin này cần đi ra cổng nào? Tất nhiên, nếu trong bảng định tuyến chưa có thông tin thì cần gọi đến hàm định tuyến, hàm này sẽ thực hiện chức năng tìm kiếm theo chiều rộng (hoặc chiều sâu) để xác định xem gói tin cần đi ra cổng nào.

**5) Xây dựng chương trình giả lập quá trình truyền tin trong mạng trung tâm dữ liệu:**

Căn cứ vào các kết quả ở trên, SV xây dựng chương trình như sau:

Với giá trị k hoặc N hoặc cả hai, SV chạy chương trình sinh ra mã nguồn NED phù hợp với topology đã giao cho mình

Với mã nguồn NED topology, cùng với mã nguồn ở các công việc khác; SV cài đặt quá trình truyền tin như sau:

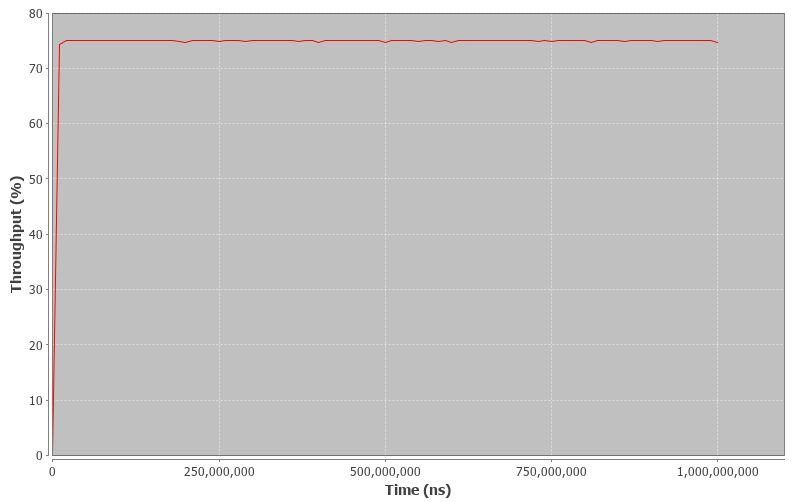
+ chọn ngẫu nhiên một nửa các nút nguồn trong số các nút nguồn của mạng.

+ Ghép cặp ngẫu nhiên của một nửa này với nửa kia. Một nút trong cặp này sẽ đảm nhận vai trò chỉ sinh ra gói tin rồi gửi, nút còn lại sẽ đảm nhận vai trò chỉ nhận gói tin rồi tính toán trong mảng lưu trữ số gói tin đã nhận của mình.

+ Các nút nguồn có vai trò sinh và gửi gói tin sẽ thực hiện việc sinh ra và gửi đi. Các nút trung gian sẽ nhận và truyền gói tin. Các nút đích sẽ nhận và tính toán.

+ kết thúc thời gian giả lập, ta sẽ có các mảng một chiều lưu trữ số các gói tin nhận được tại mỗi nút đích theo các interval time khác nhau. Ghép các mảng một chiều này để có được một ma trận hai chiều. Tiếp đó tính toán số lượng byte đã nhận trong một đơn vị thời gian, bằng cách nhân ma trận hai chiều kia với dung lượng của một gói tin rồi chia cho giá trị thời gian interval time.

+ Tính tổng tất cả các byte đã được bởi tất cả các nút đích trong từng interval time khác nhau rồi chia cho số lượng các nút đích. Cuối cùng chia cho băng thông của đường dây. Ở đây sẽ được một mảng một chiều trong đó số phần tử là số lượng các interval time, giá trị của mỗi phần tử là throughput của mạng trong một interval time. Tiếp đó vẽ đồ thị biểu diễn (cho phép dùng các công cụ để vẽ, như gnuplot), chẳng hạn như sau:



Hình 6: Thông lượng throughput theo thời gian của quá trình truyền tin trong Fat-Tree

Để dễ so sánh kết quả của SV, chương trình cần hoạt động với giá trị mặc định như sau:

+ chu kỳ sinh gói tin: 0.1ms

+ Chu kỳ hoạt động của switch: 10ns.

+ Dung lượng của gói tin: 100Kb, coi như xấp sỉ 100.000 bit.

+ Thời gian giả lập là 1s, credit\_delay là 1ns.

+ Giá trị interval time là 100ns

+ Các EXB, ENB có kích thước 5.

+ Băng thông của đường truyền là 1Gbps, coi như xấp sỉ 1 tỉ bit/s