

# Một tiếp cận thiết kế công cụ phần mềm đánh giá hiệu năng mạng liên kết kích thước lớn

## Bài báo mời

Kiều Thành Chung<sup>1,2</sup>, Nguyễn Tiến Thành<sup>2</sup>, Nguyễn Khanh Văn<sup>2</sup>

<sup>1</sup>Trường Cao đẳng nghề Công nghệ cao Hà Nội

<sup>2</sup>Trường Đại học Bách khoa Hà Nội

Tác giả liên hệ: Nguyễn Khanh Văn, vannk@soict.hust.edu.vn

Ngày nhận bài: 28/08/2019, ngày sửa chữa: 03/09/2019, ngày duyệt đăng: 08/09/2019

Xem sớm trực tuyến: 08/09/2019, định danh DOI: 10.32913/mic-ict-research-vn.v2019.n1.889

Biên tập lĩnh vực điều phối phản biện và quyết định nhận đăng: PGS.TS. Nguyễn Linh Trung

**Tóm tắt:** Bài báo này giới thiệu một cách tiếp cận riêng biệt để thiết kế công cụ phần mềm chuyên dụng cho đánh giá hiệu năng mạng liên kết kích thước lớn, nhờ đề xuất cơ chế mô phỏng giảm lược nhưng vẫn đảm bảo đánh giá được những đặc tính chính của các tô-pô và thuật toán định tuyến mới. Chúng tôi đề xuất kiến trúc tổng thể, trong đó cho phép thực hiện thí nghiệm với các tô-pô được định nghĩa sẵn hoặc tự định nghĩa, với nhiều kịch bản hay cấu hình traffic khác nhau, và hỗ trợ xây dựng tô-pô hiệu năng cao với các tính năng đánh giá cách triển khai cài đặt trên mặt sàn thực tế cho trước. Với bản phần mềm công cụ đầu tiên xây dựng trên kiến trúc đề xuất này chúng tôi đã tiến hành một số thực nghiệm với một số tô-pô mạng loại mới có kích thước hàng chục nghìn node trong khoảng thời gian ngắn khả quan, nhanh hơn khá nhiều lần so với khi tiến hành trên các công cụ mô phỏng phổ quát quen thuộc như NS3, OMNET++.

**Từ khóa:** Tô-pô mạng, hiệu năng mạng, mô phỏng mạng, mạng liên kết, công cụ mô phỏng.

---

**Title:** An Approach to Designing Software Tool Specified for Evaluating Performance of Interconnection Topologies with Large Sizes

**Abstract:** This paper introduces a special approach to designing a software tool that is specified for evaluating performance of interconnection topologies with large sizes. This is based on a proposed network simulation mechanism that is heavily reduced (much simplified from the popular network simulation standard) in order to do just enough for evaluating the performance factors related to topology and routing mechanism characteristics. We propose a general architecture for such a system that allows to experiment with pre-defined or newly defined topologies by using different experiment scripts with different traffic patterns, and also to support the designing of new high-performance topologies with features for evaluating deployment layouts. Using our initial version of this software tool we are able to perform some interesting experimental evaluation of a few modern topologies with sizes as large as tens of thousands in very encouragingly short periods of time, multiple times faster than performed in popular general simulators such as NS3 or OMNET++.

**Keywords:** Network topology, network performance, network simulation, interconnection network, simulation software.

---

## I. GIỚI THIỆU VẤN ĐỀ NGHIÊN CỨU

Nghiên cứu tô-pô<sup>1</sup> của các mạng liên kết (MLK: Interconnection Network) [1, 2] có vai trò quan trọng trong lĩnh vực tính toán song song và kiến trúc mạng máy tính. Một giải pháp tô-pô hiệu quả sẽ đóng góp quyết định vào hiệu năng của các mạng lưới tính toán lớn như các hệ máy tính hiệu năng cao (HPC: High-Performance Computer) [2] hay trung tâm dữ liệu hiện đại (TTDL: Data Center) đang ngày càng trở thành nhu cầu cấp thiết nhằm phục vụ các dịch vụ tính toán lớn.

Một thách thức hiện nay đối với giới chuyên môn là thiết kế các hệ thống MLK hiệu năng cao trong TTDL với kích

thước lớn, có thể lên đến hàng trăm nghìn máy. Bản thân việc tổ chức thực nghiệm để đánh giá hiệu năng một tô-pô mạng (network topology) đề xuất mới đã là một vấn đề không tầm thường vì hầu hết các công cụ mô phỏng mạng phổ quát hiện nay chỉ đáp ứng cho kích thước mạng lên đến hàng ngàn nút.

Với sự phát triển vũ bão của các ứng dụng và dịch vụ Internet, của công nghệ Dữ Liệu Lớn (Big Data), các trung tâm dữ liệu hiện đại đang được quan tâm đầu tư nhiều, trong đó một yêu cầu đặc thù phục vụ cho xu hướng mới này là thiết kế MLK nhằm đảm bảo tốt tính mềm dẻo (flexibility) và khả năng mở rộng (scalability). Cụ thể là việc bổ sung thêm các nút (node) mạng, tăng/giảm kích thước mạng có thể thực hiện nhiều lần với qui mô đa dạng, nhưng vẫn phải đảm bảo hiệu năng mạng.

<sup>1</sup>Topology: Cấu trúc tổ chức hình học của mạng máy tính.

Với kích thước mạng càng ngày càng lớn, các vấn đề về tiết kiệm chi phí thiết bị và cáp, tiết kiệm năng lượng khi tải thấp, v.v. trở thành những chủ đề rất đáng quan tâm. Có thể thấy điều này trong hàng loạt nghiên cứu trong giai đoạn gần đây về thiết kế MLK trong TTDL HELIOS [3], BCUBE [4], DCELL [5], Scafida [6], MD-Cube [7], VL2 [8]. Vì vậy các dạng tô-pô truyền thống trở nên phần nào lạc hậu, giới nghiên cứu đang tích cực đề xuất những tiếp cận, những dạng kiến trúc tô-pô mới phù hợp hơn cho các yêu cầu hiện đại, như Smallworld Data-Center [9], JellyFish [10], Space Shuffle [11], FatTree [12], SKYWALK [13], Dragonfly [14].

Một trong những thách thức lớn của địa hạt nghiên cứu này cũng là vấn đề đã đề cập trên: làm sao tổ chức đánh giá qua thực nghiệm với một mô hình mạng kích thước lớn, có thể vượt xa các kích thước mạng truyền thống. Để đánh giá một tô-pô MLK, bên cạnh các phương pháp đánh giá dựa trên cài đặt các thuật toán chuyên dụng để tính toán các đặc trưng hiệu năng (xoay quanh phân tích đồ thị), thực nghiệm qua công cụ mô phỏng mạng được coi là yếu tố có tiếng nói quyết định. Tuy nhiên việc thực hiện cài đặt các mạng kích thước lớn đến hàng chục nghìn nút tính toán trên các công cụ mô phỏng mạng hay hệ phân tán phổ biến hiện nay như NS3 [15], OMNET++ [16], SIMGRID [17], được xem là bất khả thi. Thậm chí với cấu hình máy tính tân tiến hàng đầu, cài đặt mạng lên đến 1000 nút đã là khá khó khăn<sup>2</sup>. Khó khăn chủ yếu là do mỗi nút tính toán trong các công cụ mô phỏng mạng phổ quát này được thiết kế để mô phỏng đầy đủ các tầng giao thức như các kiến trúc phổ biến (mô hình OSI bảy tầng hay mô hình TCP/IP). Khối lượng xử lý lớn ở mỗi nút sẽ giới hạn kích thước tập hợp các đối tượng mạng có thể cài đặt trong hệ mô phỏng.

Tuy nhiên chúng tôi nhận thấy rằng với mô hình MLK, các nhà nghiên cứu chỉ quan tâm tới các yếu tố hiệu năng xoay quanh việc thực hiện các nhiệm vụ tầng giao vận và ứng dụng khi tải lớn chứ không quan tâm thực sự tới các tầng thấp. Quan sát này mở ra một tiếp cận mới là có thể cải tiến lõi hệ phần mềm mô phỏng nhằm hướng tới mạng kích thước lớn hơn thông qua việc rút gọn giản lược các chức năng của các tầng dưới (tầng mạng trở xuống). Thậm chí các quá trình truyền tin có thể thay thế bằng các mô hình ngẫu nhiên tương ứng (mô phỏng lõi đường truyền) để sinh độ trễ phù hợp. Việc giản lược các tầng dưới đương nhiên là không phù hợp cho mô phỏng một quá trình hoạt động đầy đủ của một hệ thống mạng trong thực tế, nhưng lại không thực sự ảnh hưởng tới việc so sánh đánh giá giữa các tô-pô mạng với nhau theo các tiêu chí hiệu năng cơ bản.

<sup>2</sup>Mỗi lần thực hiện một thực nghiệm theo một kịch bản mô phỏng cơ bản có thể là hàng giờ trong khi cần thực hiện hàng trăm thực nghiệm để so sánh đánh giá một số tô-pô mạng với nhiều chế độ hoạt động và khai thác.

Trong bài báo này chúng tôi đề xuất một tiếp cận mới: thiết kế phần mềm đánh giá tô-pô MLK thông qua thực nghiệm có mô phỏng, trong đó tập trung giản lược hóa các tầng thấp (chủ yếu là địa bàn để xây dựng các thuật toán định tuyến mới khi cần) và tập trung vào tổ chức các thí nghiệm chuyên dụng cho MLK ở các tầng trên (giao vận và ứng dụng). Thông qua tiếp cận mới này, chúng ta có thể đánh giá một đề xuất mới bằng việc so sánh với các giải pháp đã có trên những cấu hình mạng kích thước lớn mà trước kia chưa thể thực hiện được.

Bài báo này trình bày những nét chính của tiếp cận mới trên và đề xuất một kiến trúc cơ sở cho hệ công cụ phần mềm đánh giá tô-pô MLK theo hướng này. Dựa vào kiến trúc cơ sở này, chúng tôi xây dựng những mô-đun nền móng đầu tiên, cho phép thực hiện các đánh giá so sánh tô-pô trên các tiêu chí hiệu năng cơ bản liên quan đến thông số đồ thị và định tuyến. Các đóng góp cụ thể như sau:

- Đề xuất thiết kế tổng thể công cụ đánh giá hiệu năng MLK, kết hợp tính toán theo mô hình và mô phỏng giản lược (SSiNET). Công cụ này cho phép người dùng tạo các tô-pô mạng mới, cài đặt với các tham số cho trước (kích thước lớn được hỗ trợ cơ chế tự động sinh theo mẫu) thử nghiệm với các thuật toán định tuyến có sẵn và cài mới. Từ đó đánh giá hiệu năng theo các tiêu chí cơ bản và nâng cao, hỗ trợ thiết kế mặt sàn (layout) cho trước và tính toán giá thành.
- Thực hiện các mô-đun cơ bản cho phép tính toán các thông số đồ thị và định tuyến.
- Tạo cơ chế cho việc cài đặt thuật toán định tuyến và cài đặt sẵn một số tô-pô đã biết (ví dụ: 2-D Torus [9], Fat-tree [12], Kleinberg Small-world [9]) và thuật toán định tuyến đã biết (ví dụ: Shortest Path Routing [1], Two-level Table [12], Compact Routing [18]).
- Với bản phần mềm công cụ đầu tiên, tiến hành thực nghiệm để đánh giá so sánh một số tô-pô mới đang được quan tâm và thuật toán định tuyến tương ứng để đánh giá độ trễ truyền tin (latency).

Nội dung chính được tổ chức như sau. Mục II trình bày các nghiên cứu liên quan. Mục III đề xuất thiết kế SSiNET. Mục IV nghiên cứu đánh giá thử nghiệm.

## II. CÁC NGHIÊN CỨU LIÊN QUAN

Để nghiên cứu và thiết kế một MLK hiệu quả, các nhà nghiên cứu tập trung vào hai vấn đề chính: (i) thiết kế tô-pô mạng; (ii) xây dựng thuật toán định tuyến, để khai thác tối đa các tính chất đặc thù của tô-pô tương ứng.

Các cách thực hiện có thể theo ba hình thức như sau: (i) chỉ xây dựng tô-pô và sử dụng các thuật toán định tuyến đã tồn tại; (ii) chỉ xây dựng thuật toán định tuyến cho các tô-pô đã biết; (iii) xây dựng mới cả tô-pô và thuật toán định tuyến tương ứng.

## 1. Tô-pô và thuật toán định tuyến

Một tô-pô mạng, tức cấu trúc hình học của mạng, được biểu diễn bằng đồ thị  $G = (V, E)$ , mỗi thiết bị chuyển mạch (switch) hoặc máy tính toán (host) trong mạng máy tính thực tế được biểu diễn bởi một nút mạng (node) trong đồ thị  $G$ . Tập hợp các switch hoặc host trong mạng được biểu diễn bởi tập đỉnh  $V$  trong đồ thị  $G$  và tập hợp các liên kết (link) giữa các switch hoặc host được biểu diễn bằng tập cạnh  $E$ .

Các tô-pô và thuật toán định tuyến (RA: Routing Algorithm) đã được nghiên cứu từ lâu, nhiều tô-pô trong số đó đã được triển khai áp dụng phổ biến trên thực tế. Ví dụ, tô-pô siêu khối cơ sở  $k$ , số chiều  $n$  ( $k$ -ary  $n$ -cube hypercube) định tuyến theo giao thức Duato [1] được sử dụng trong các siêu máy tính như là BlueGene/L Anton-2 hoặc Cray XT5 [19]. Gần đây, các tô-pô truyền thống được xem là không đáp ứng được các tiêu chí hiệu năng hiện đại như độ trễ truyền tin thấp và tính cơ giãn qui mô.

Vì vậy như là một tiếp cận mới, thiết kế tô-pô mạng dựa trên mô hình đồ thị ngẫu nhiên [9] đã được quan tâm và đem lại nhiều cải tiến hiệu năng nổi bật. Tô-pô ngẫu nhiên (RSN: Random Shortcut Network), được cấu trúc dựa trên một tô-pô cơ bản như là Mesh, Torus, Hypercube, Flattened Buttery, v.v. và bổ sung thêm các liên kết dài giữa các nút ở xa nhau. Các liên kết dài được tạo ra bởi việc sử dụng một phân bố xác suất ngẫu nhiên nào đó. Các RSN đã được công bố như là Jellyfish [10], Small-world [9].

Một số tiêu chí cơ bản của hiệu năng mạng được đánh giá dựa trên việc tính toán các tham số chính như đường kính mạng (diameter), độ dài trung bình đường định tuyến (ARPL: Average Routing Path Length), độ trễ truyền tin [1], kích thước bảng định tuyến (RTS: Routing Table Size). Độ trễ truyền tin là khoảng thời gian gói tin bắt đầu được khởi tạo tại nút nguồn (source node), gửi đi trên kênh truyền và nhận được tại nút đích (destination node). Độ trễ đạt giá trị thấp là dấu hiệu của hiệu năng cao. Độ trễ lý thuyết tối thiểu có thể được tính toán thông qua mô hình hóa mạng như một cặp đồ thị và thuật toán định tuyến cho trước.

Định tuyến là quá trình định hướng cho một gói tin di chuyển từ nút nguồn  $S$  đến nút đích  $T$  trong mạng. Các thuật toán định tuyến lựa chọn và xác định đường đi (path) để gói tin truyền từ nguồn tới đích trong một MLK xác định. Thuật toán định tuyến đường đi ngắn nhất (SPR: Shortest Path Routing, còn gọi là Minimal Routing) [1] được sử dụng trong các mạng dạng chuẩn tắc (regular). Minimal Routing đảm bảo yếu tố stretch- $1^3$  [20]. Ví dụ: 2-D Torus, 3-D Torus [1], FatTree [12], Hypercube sử dụng SPR. Các thuật toán định tuyến tùy chọn (Custom Routing) được thiết

<sup>3</sup>Tỉ lệ giữa chiều dài đường định tuyến với chiều dài ngắn nhất trên tô-pô mạng giữa cặp đỉnh bất kỳ trong mạng.

kế riêng cho từng loại tô-pô cụ thể (còn gọi là Non-Minimal Routing), với stretch-3 [18, 21], thông thường là các mạng dạng không chuẩn tắc (irregular). RSN sử dụng thuật toán SPR hoặc TZ (Thorup & Zwick [18]), JellyFish [10] sử dụng  $k$ -SPR.

## 2. Công cụ mô phỏng mạng

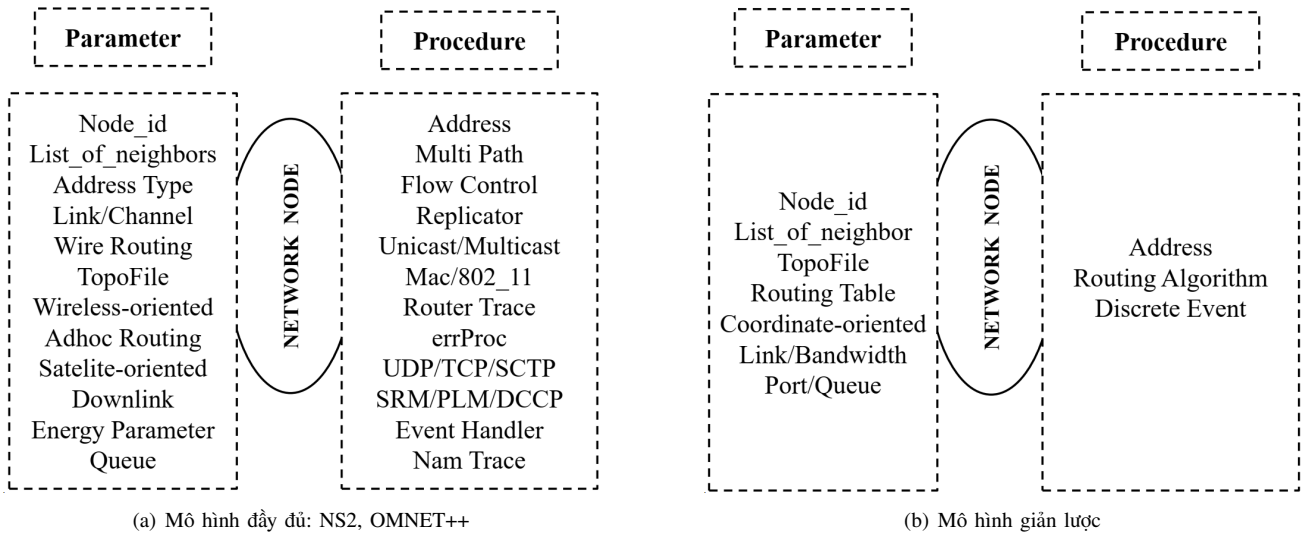
Hiện nay các nhà nghiên cứu thường tự phát triển phần mềm tính toán đặc trưng và đánh giá các tô-pô MLK (và thuật toán định tuyến), và khi cần thực hiện thực nghiệm mô phỏng có thể sử dụng các phần mềm mô phỏng mạng phổ quát, phổ biến hiện nay như NS2 [22], NS3 [15], OMNET++ [16], SIMGRID [17].

Hệ công cụ NS2 [22] thường được sử dụng thực nghiệm đối với các mạng không dây. NS2 có nhược điểm là quá trình mô hình hóa mạng khá phức tạp và tốn nhiều thời gian và bộ nhớ. Ngoài ra, nó cũng không định nghĩa sẵn các thiết bị như switch, router và không có giao diện người sử dụng (GUI: Graphic User Interface) để hiển thị kết quả. Là thế hệ đời sau, NS3 [15] cũng mô phỏng hệ thống bằng cách cấu hình các node mạng một cách đầy đủ, do đó NS3 cũng chỉ thực thi được với mạng kích thước nhỏ từ vài trăm tới nghìn nút.

OMNET++ [16] hỗ trợ mô phỏng song song thông qua việc sử dụng các thư viện MPI (Message Passing Interface), thực hiện mô phỏng lưu lượng trong mạng, tạo mô hình các giao thức, xây dựng mô hình các hàng đợi trong mạng, xây dựng các bộ vi xử lý đa nhiệm và các hệ thống phân tán khác, kiểm tra lại kiến trúc phần cứng, đánh giá khía cạnh về hiệu suất của các hệ thống phần mềm phức tạp.

SIMGRID [17] là một công cụ cung cấp các chức năng cơ bản để mô phỏng các ứng dụng phân tán quy mô lớn như Grid, P2P hay Cloud. Công cụ mô phỏng SIMGRID có thể được sử dụng để đánh giá thuật toán định tuyến, các ứng dụng MPI. Kích thước mạng được sử dụng để đánh giá chỉ đạt khoảng 256 nút đối với máy tính toán thông thường và đạt 1024 đối với máy tính toán có cấu hình mạnh.

Nhìn chung, các phần mềm trên có tính phổ quát cao, tập trung mô phỏng các thiết bị và các giao thức mạng một cách chi tiết và đầy đủ. Nhằm đảm bảo phản ánh tương đối trọn vẹn các quá trình và yếu tố kỹ thuật đa dạng trong thực tiễn, các hệ này tập trung mô phỏng chi tiết. Lớp mạng với thư viện cấu hình đối tượng phân cấp phức tạp, mô phỏng rất sát thực tế các quá trình truyền tin. Do vậy, lượng thông tin được lưu trữ và khối lượng tính toán trở nên đồ sộ và thường khó thực thi được với mạng kích thước lớn (thậm chí là rất khó khăn khi kích thước đạt đến hàng nghìn nút). Trong khi đó, việc khảo sát đánh giá các MLK thường chỉ đòi hỏi thực nghiệm đánh giá các đặc trưng đồ thị (cấu hình mạng) và tính toán các hiệu năng giải thuật định tuyến, và



Hình 1. Cấu trúc node mạng.

nếu sử dụng mô phỏng giao thức thì chỉ là mức tương đối đơn giản nhằm đánh giá một vài đặc trưng tổng thể như thông lượng mạng.

### III. CÔNG CỤ PHẦN MỀM SO SÁNH VÀ ĐÁNH GIÁ TÔ-PÔ MLK SSINET

#### 1. Kiến trúc tổng quan

##### 1) Ý tưởng cơ bản:

Như đã đề cập, các hệ công cụ mô phỏng phổ quát là không khả thi khi đưa vào mô phỏng những mạng kích thước tương đối lớn, vì các đối tượng được xây dựng phức tạp. Hình 1(a) cho minh họa một đối tượng nút mạng với các thuộc tính chi tiết và các phương thức truyền tin, kết nối mạng.

Nhằm mở rộng khả thi việc đánh giá các tô-pô mạng lớn (có thể hàng chục nghìn, hay lên đến trăm nghìn nút), phương pháp của chúng tôi là tìm cách giản lược cấu trúc thông tin của đối tượng nút mạng và các phương thức hoạt động cơ bản, chỉ giữ lại các yếu tố chủ chốt cần thiết. Tất nhiên, hệ công cụ giản lược sẽ không còn khả năng đáp ứng các yêu cầu đa dạng của mô phỏng mạng, nhưng tập trung vừa đủ vào các thành phần cần thiết để vẫn không ảnh hưởng đến việc đánh giá các tiêu chí hiệu năng cơ bản của MLK.

Các hình 1(a) và 1(b) liệt kê và so sánh các thuộc tính cấu hình nút mạng trong NS2 [22] và cấu hình nút mạng do chúng tôi đề xuất. Theo tiếp cận mô phỏng giản lược này, chúng tôi nhận thấy để thỏa mãn đảm bảo việc đánh giá hiệu năng cơ bản của các tô-pô MLK, một hệ mô phỏng giản lược chỉ cần tập trung vào thực hiện các yếu tố cốt lõi sau đây.

- *Cấu trúc thông tin đồ thị và thuộc tính vật lý:* Thể hiện được sự liên kết giữa các nút mạng, tức cấu trúc thông tin dạng đồ thị, và các thuộc tính cơ bản về mặt vật lý và không gian của các nút và đối tượng liên quan (năng lượng, tốc độ, băng thông đường truyền, độ dài cáp nối. v.v).
- *Thuật toán và cấu trúc thông tin định tuyến:* thể hiện được khả năng khai thác liên kết mạng để đảm nhiệm các nhiệm vụ định tuyến và truyền tin theo các thuật toán và giao thức được thiết lập sẵn.
- *Mô phỏng hoạt động truyền tin mạng cơ bản:* thể hiện thu phát gói tin theo mô hình thiết lập trước, thiết lập lịch trình sự kiện (discrete event scheduler), cơ chế nhật ký và điều khiển thời gian tương đối (relative timing).
- *Mô phỏng hoạt động tương tác và cạnh tranh tài nguyên giữa các nút mạng:* thể hiện hiện tượng tranh tài nguyên khi có số lượng nhiệm vụ truyền tin đáng kể (traffic mạng lớn và phức tạp), dòng xếp hàng gói tin phát hoặc nhận chờ xử lý.

Có thể nói với tiếp cận mô phỏng giản lược này, chúng tôi chỉ quan tâm đánh giá và so sánh các MLK như là bao gồm hai yếu tố: thành phần cơ bản đồ thị tô-pô và hoạt động truyền tin theo thuật toán định tuyến cho trước. Do đó chúng tôi thực thi mô phỏng với các điều kiện lý tưởng, như: truyền tin không lỗi, không xử lý sửa lỗi truyền tin, không xảy ra nhiễu trên mạng, không tiến hành gửi lại gói tin và thời gian thiết lập kênh truyền không đáng kể. Hệ công cụ tập trung vào trọng tâm chính là mô tả phản ứng của các đối tượng mạng khi thay đổi phương thức định tuyến và đánh giá hiệu năng mạng tương ứng. Khi thay đổi thuật toán định tuyến, chúng tôi giám sát cơ chế biến đổi

trạng thái của các đối tượng (các nút mạng) và việc lưu trữ thông tin định tuyến tại mỗi node mạng, cập nhật trạng thái toàn mạng. Chính do những yếu tố rút gọn lý tưởng này nên chúng tôi tạm gọi phương pháp tiếp cận này là *mô phỏng giản lược*. Có thể nói phương pháp thực nghiệm này sẽ thua kém mô phỏng đầy đủ khi chúng ta cần khảo sát chi tiết việc vận hành trong một điều kiện thực tế giả định nào đó, tuy nhiên sẽ vẫn đảm bảo hỗ trợ tốt việc đánh giá so sánh giữa các giải pháp, ví dụ như so sánh hiệu năng giữa hai thuật toán định tuyến khác nhau hay hai thiết kế tô-pô khác nhau, v.v.

Sau đây chúng tôi đi sâu trình bày về đề xuất kiến trúc tổng thể của hệ công cụ thực nghiệm hỗ trợ đánh giá hiệu năng MLK đang được xây dựng theo cách tiếp cận đã phân tích trên.

Hệ công cụ này được chúng tôi đặt tên ngắn gọn là **SSiNET** (Simplified Simulation of the interconnection **NET**work) với mục tiêu chung là cung cấp một phần mềm chuyên dụng cho việc thiết lập, so sánh và đánh giá có thông qua mô phỏng giản lược đối với các tô-pô MLK có kích thước lớn.

## 2) Kiến trúc mô-đun chức năng và giao diện của SSiNET:

Hệ SSiNET có ba khối chức năng chính. Hai khối chức năng đầu xác lập các nền tảng thỏa mãn được bốn yếu tố thành phần chính trong mô phỏng giản lược mà chúng tôi đã nêu trên. Khối chức năng thứ ba được xây dựng như một tầng trên để người sử dụng trực tiếp định nghĩa, xác lập và tổ chức một cấu hình tô-pô mạng phức tạp và tiến hành các thực nghiệm đánh giá theo các dạng/thể loại thí nghiệm đã được hỗ trợ xác lập trước.

**Khối chức năng mô tả cấu trúc và thiết lập cấu hình mạng:** cho phép đảm bảo các thao tác thiết lập đối tượng nút và cấu trúc mạng cơ bản (tức là cấu trúc thông tin đồ thị và thuộc tính vật lý) với các chức năng sau đây.

- Tạo tô-pô một cách linh hoạt, với kích thước mạng lớn (lên tới hàng chục nghìn nút hoặc hơn nữa).
- Cung cấp thư viện thiết lập sẵn gồm các cấu trúc tô-pô cơ bản như Ring, Grid, Torus, Mesh.
- Hỗ trợ người sử dụng (NSD), là nhà nghiên cứu, để có thể khai báo và tạo ra các tô-pô mạng mới theo một quy luật nào đó.

**Khối chức năng cài đặt định tuyến và mô phỏng giản lược:** thỏa mãn ba yếu tố còn lại trong bốn nêu trên nhờ đó xác lập cơ chế định tuyến và mô phỏng giản lược. Các chức năng cụ thể là:

- Thiết lập sẵn một số thuật toán định tuyến cơ bản (như Minimal Routing).
- Hỗ trợ xây dựng các thuật toán tự định nghĩa (Custom Routing) để áp dụng cho một tô-pô cụ thể.

- Mô phỏng quá trình thu phát và xử lý gói tin.
- Bộ lập lịch và quản lý sự kiện rời rạc (Discrete Event Scheduler).
- Kiểm soát tương tranh kênh truyền và quản lý dòng xếp hàng.

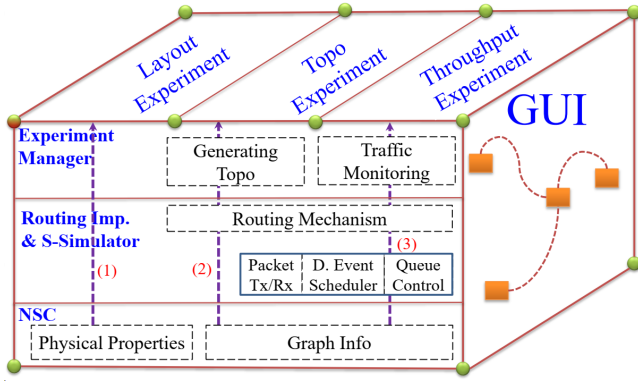
### Khối chức năng tổ chức quản lý thực nghiệm:

- Quản lý thực nghiệm về phân tích đồ thị (graph analysis): cung cấp công cụ để NSD xây dựng các bài thí nghiệm thực hiện đánh giá các tiêu chí hiệu năng liên quan đến đồ thị (như đường kính mạng, độ dài đường đi định tuyến trung bình, độ trễ truyền tin theo mô hình).
- Quản lý thực nghiệm về giao thông (traffic-related): cung cấp công cụ để xây dựng các bài thực nghiệm đánh giá về sức tải (capacity) và thông lượng (throughput) mạng.
- Quản lý thực nghiệm về triển khai thiết bị mạng trên mặt sàn (layout design support) và đánh giá giá thành (cost evaluation).

Cấu trúc ba khối chức năng cơ bản của SSiNET được minh họa tại hình 2. Tại đây chúng tôi chọn thể hiện minh họa bằng một cấu trúc tổ chức ba chiều để làm rõ thêm các mặt quan trọng khác của hệ thống tổng thể.

Ở mặt chính chúng tôi minh họa khái quát ba khối chức năng chính của hệ thống: mô tả cấu trúc và thiết lập cấu hình mạng (NSC: Network Structure and Configuration), cài đặt định tuyến và mô phỏng giản lược (RI&SS: Routing Implementation and Simplified Simulation) và tổ chức quản lý thực nghiệm (EM: Experiment Manager). Có thể thấy kiến trúc này phần nào thể hiện tư duy phân tầng thường thấy: khi đối chiếu với mô hình OSI bảy tầng thì NSC sẽ giữ vai trò mô phỏng tầng 1 (physical), RI&SS mô phỏng “giản lược” các tầng từ 2 đến 4 (data link, network, transport). Còn lại, EM là tương ứng với tầng 7 (application) hỗ trợ NSD khai thác hệ công cụ (theo các bài thực nghiệm chuyên dụng).

Ở mặt trên, chúng tôi thể hiện ba khía cạnh cơ bản trong việc khai thác hệ công cụ để đánh giá (và so sánh) các tô-pô MLK. Như đã biết, nói đến hiệu năng mạng (network performance) là nói đến hai loại tiêu chí hiệu năng chính, trong đó loại thứ nhất là theo phương pháp phân tích đồ thị như: đường kính đồ thị, ARPL, độ trễ truyền tin. Loại tiêu chí thứ hai là thể hiện các hiệu năng mạng trong hoạt động có tải (traffic, with load) thường được thực nghiệm bằng phương pháp mô phỏng như: thông lượng, công suất tiêu thụ (power consumption), sức tải. Chúng tôi cũng mở rộng thêm một khía cạnh thứ ba nữa, bắt đầu được giới nghiên cứu quan tâm gần đây, đó là về hiệu quả bố trí triển khai mặt sàn (layout deployment) và giá thành. Một cách tóm tắt, chúng tôi đưa ra ba dạng thực nghiệm chính mà hệ thống hỗ trợ: topology (cho phân tích đồ thị), throughput



Hình 2. Sơ đồ thiết kế tổng quan.

(cho đánh giá hiệu năng khi có tải) và layout (cho đánh giá bố trí mặt sàn và triển khai). Dưới đây phân tích cụ thể hơn về từng loại thực nghiệm.

**Layout Experiments:** Thực nghiệm về bố trí và sắp xếp các nút mạng trên sàn với dạng khai báo, là mặt sàn thực tế lắp đặt các nút trong server-room. Lớp thực nghiệm và đánh giá này nảy sinh từ thực tiễn xây dựng TTDL cỡ vừa và nhỏ trên các mặt sàn không chuẩn (không gần nhau, ví dụ như nhiều phòng trong cùng một tòa nhà), mà một vấn đề kỹ thuật quan trọng thường đưa ra là làm thế nào để áp dụng tô-pô đã chọn trên các mặt sàn đó. Như minh họa bởi đường nét đứt số (1) trong hình 2, phân hệ Layout Experiment được thực thi dựa trên các quan hệ liên kết giữa các nút (trên đồ thị), và các thông số tọa độ trên mặt sàn, ta cần tính toán phương án cài đặt để các thiết bị mạng được bố trí phù hợp trên một diện tích cụ thể (Physical Properties, được tạo ra trong khối chức năng NSC). Đôi khi có thể có yêu cầu bố trí trên nhiều mặt sàn khác nhau, ví dụ: đầu vào là tập các sàn  $L_i$  với cấu trúc hình học và diện tích của sàn thứ  $i$  cho trước. Đây là lớp bài toán mà chúng tôi mong muốn nghiên cứu mở rộng sau này.

**Topo Experiments:** Thực nghiệm nhằm đánh giá bằng phương pháp phân tích đồ thị để đưa ra các thông số hiệu năng mạng theo nhóm này. Phương pháp này không đòi hỏi thực sự tiến hành mô phỏng truyền tin giữa các cặp node trong mạng, mà ở chế độ không tải (zero-load), chủ yếu xác định và tính toán độ dài đường đi giữa mọi cặp nút dựa trên cấu trúc tô-pô và cơ chế định tuyến đã xác lập (thuật toán định tuyến đã chọn). Trong hình 2, chúng tôi minh họa sự liên kết trực tiếp giữa mô-đun phần mềm này với các mô-đun tương ứng ở tầng dưới thông qua đường nét đứt số (2). Các thực nghiệm tô-pô cần sử dụng các thông tin đồ thị (Graph Info) trong khối chức năng NSC và được áp dụng thuật toán định tuyến, thông tin bảng định tuyến (Routing Mechanism) trong khối chức năng RI&SS.

**Throughput Experiments:** Thực nghiệm để tính thông lượng và các yếu tố sức tải, đánh giá hiệu năng mạng

trong chế độ có tải. Trong thực nghiệm này, các gói tin sẽ được gửi đồng loạt giữa mọi cặp nút  $(s, t)$  bất kỳ trong một chu kỳ thời gian. Đường nét đứt số (3) thể hiện sự liên kết giữa các khối chức năng để thực nghiệm thông lượng. Các thông tin đồ thị được sử dụng cùng với thông tin bảng định tuyến. Các tham số như số gói tin gửi/nhận (Tx/Rx packet), các sự kiện và cơ chế quản lý hàng đợi gói tin (queue control) được cài đặt để tiến hành gửi giữa các cặp nguồn-đích. Thực nghiệm thông lượng sẽ đánh giá tỉ lệ giữa số lượng gói tin được gửi đi từ tập nguồn  $S$  ( $s \in S$ ) và số lượng gói tin nhận được tại tập đích  $T$  ( $t \in T$ ) trong một chu kỳ thời gian quan sát. Khi thực hiện liên tiếp nhiều thí nghiệm với yêu cầu giao thông tăng dần, thông số này sẽ hội tụ đến giá trị cực đại (thời điểm bão hòa), qua đó ta đánh giá sức tải của mạng.

Ở chiều thứ ba (mặt bên của khối lập phương ở hình 2) chúng tôi chỉ nhằm nêu bổ sung một khía cạnh đơn giản mà không đi sâu. Đó là, hệ công cụ cần có hỗ trợ GUI để biểu đạt và diễn giải nhiều dạng thông tin và kết quả chuyên biệt, như: biểu diễn đồ thị (cỡ lớn cần có qui hoạch phân lớp với cơ chế thu phóng thích hợp), biểu diễn bố trí mặt sàn, biểu diễn traffic và thông lượng.

## 2. Thiết kế chi tiết

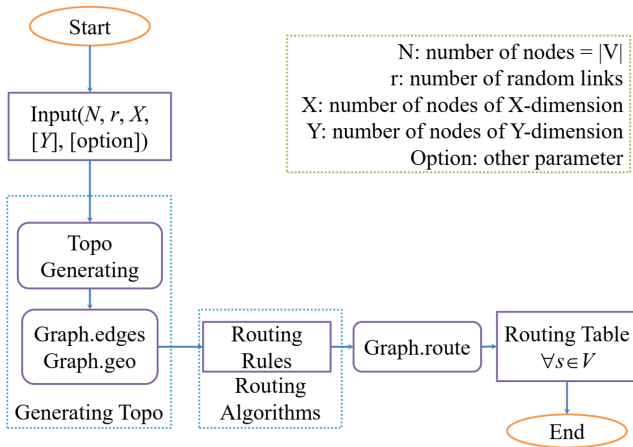
Dưới đây chúng tôi sẽ trình bày bổ sung thêm chi tiết thiết kế của một số thành phần quan trọng.

### 1) Khối chức năng mô tả cấu trúc và thiết lập cấu hình mạng:

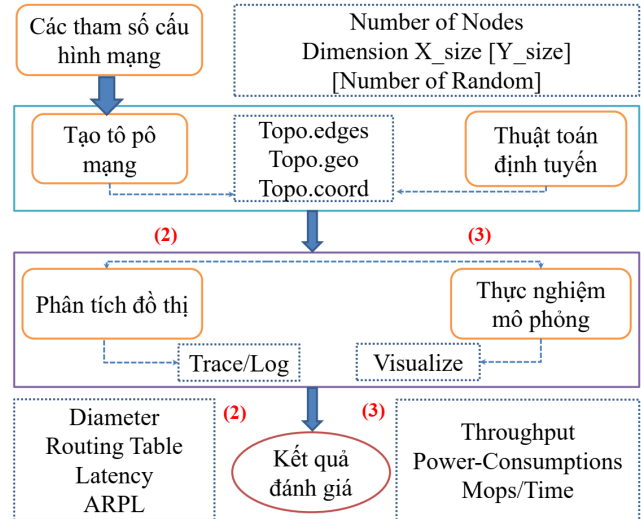
Khối chức năng này có nhiệm vụ hỗ trợ NSD để có thể nhanh chóng dễ dàng khai báo và mô tả cấu trúc đồ thị của tô-pô cần khảo sát và các thông số vật lý của cấu trúc nút tính toán. Dựa vào các tham số cấu trúc tô-pô, phần mềm sẽ tạo ra tô-pô được lưu trữ dưới dạng tệp tin *Graph.edges* (Graph Info) lưu trữ liên kết giữa các nút mạng và *Graph.geo* (Physical Properties) lưu trữ các tọa độ của các nút trong mạng.

Cấu trúc đồ thị của tô-pô mạng, ký hiệu  $G = (V, E)$ , có thể được xác định với kích thước mạng  $N = |V|$ ,  $X\_size$  và  $Y\_size$  xác định kích thước của mỗi chiều của đồ thị  $G$  (trong hầu hết trường hợp thực tiễn, các nút tính toán được bố trí như các đỉnh nút của một lưới vuông trên thực địa). Tô-pô theo tiếp cận mạng ngẫu nhiên sẽ có thêm tham số *Number\_of\_random* xác định số lượng liên kết ngẫu nhiên tại mỗi nút mạng. Cấu trúc đồ thị được chọn trong một tập thư viện cài đặt sẵn, hoặc được xây dựng mới với một số điều kiện cần thỏa mãn.

Là một thành phần cơ bản của khối chức năng, mô-đun chức năng tạo tô-pô (Topo Generating) sẽ hỗ trợ việc xây dựng một tô-pô có kích thước lớn thông qua việc khai báo kiểu dạng và các tham số cơ bản. Đầu vào của chức năng



Hình 3. Lưu đồ tạo tô-pô mạng



Hình 4. Sơ đồ thiết kế chi tiết.

này bao gồm các tham số như đã được đề cập trong mục III.1.1. Dựa trên tính chất đặc trưng của mỗi loại tô-pô (ví dụ như dạng lưới, dạng ngẫu nhiên, dạng có cấu trúc) hệ thống tạo ra dữ liệu cho tô-pô đó; dữ liệu này được lưu trữ tại tệp tin *Graph.edges* (lưu trữ danh sách các cạnh kề của mọi nút trong đồ thị), tệp tin *Graph.geo* (lưu trữ vị trí của mọi nút trong đồ thị) và bảng định tuyến ( $s \in S$ ) được lưu trữ tại mỗi nút trong đồ thị. Hình 3 là sơ đồ chi tiết của quá trình khai báo và sinh dữ liệu tô-pô này.

### 2) Các quá trình xử lý chính:

Hình 4 mô tả sơ đồ các luồng điều khiển và quá trình xử lý chính trong SSiNET. Về cơ bản để so sánh và đánh giá các tô-pô, chúng ta có hai loại thực nghiệm chính, thực nghiệm phân tích đồ thị (Topo Experiment), trong đó các thông số dựa trên đồ thị tô-pô được đánh giá mà không cần quan tâm đến traffic (không tải) và thực nghiệm có traffic (Throughput Experiment) trong đó ta mong muốn đánh giá sức tải của MLK đối với những mẫu traffic khác nhau. Quá trình xử lý phân tích đồ thị được thể hiện qua nửa trái với ký hiệu (2) trong hình vẽ, còn quá trình đánh giá sức tải được thể hiện qua nửa phải với ký hiệu (3). Cả hai quá trình này đều khai thác đầy đủ phân hệ NSC.

Phần xây dựng cơ chế định tuyến là một cấu thành đặc biệt mà về hình thức thì có thể xem như nằm trong NSC vì nó cho phép NSD đăng ký lựa chọn thuật toán đã có sẵn trong thư viện và khai báo các thông số cần thiết. Tuy nhiên cấu thành này cũng có cho phép NSD xây dựng thuật toán định tuyến riêng của mình (theo mẫu và hướng dẫn) và bộ API theo chuẩn qui định rồi đưa vào cài đặt trong hệ thống. Tất nhiên, về bản chất sự cài đặt bổ sung các thuật toán định tuyến mới này là ở tầng 2, RI&SS. Qua đó, khi một thuật toán định tuyến được chọn sử dụng, nó sẽ được áp dụng với tô-pô được tạo ra để chuẩn bị cho quá trình thực nghiệm mô phỏng.

Lưu ý thêm là nhãn (2) trong hình 3 tương ứng với đường số (2) trong hình 2, thể hiện quá trình thực nghiệm Topo Experiment để cho ra đánh giá các thông số hiệu năng mạng như đường kính, độ trễ truyền tin, ARPL, RTS. Nhãn (3) tương ứng với đường số (3) trong hình 2 thể hiện quá trình Throughput Experiment để đánh giá sức tải và thông lượng của mạng.

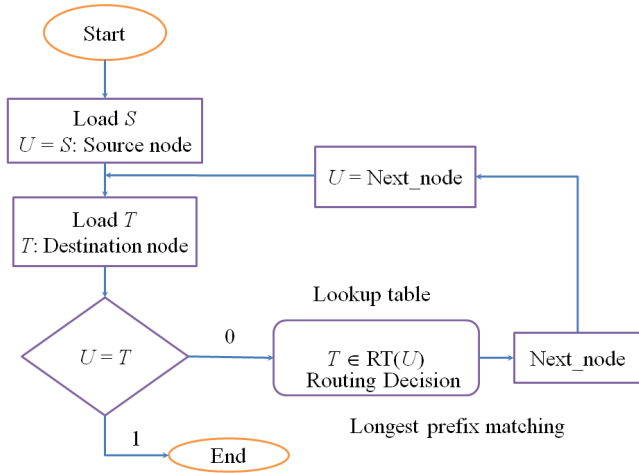
### 3) Xây dựng cơ chế định tuyến:

Mô-đun này xác định thông tin định tuyến lưu trữ tại mỗi nút mạng và được thực hiện tại mỗi nút này (phân tán). Tùy thuộc vào quy tắc định tuyến (routing rule) được áp dụng, chương trình sẽ cập nhật thông tin định tuyến tại các nút khác trong bảng định tuyến của mỗi nút ( $RS_{u \in V}$ ), sử dụng các phương thức như CAM [23] hay TCAM [24]. Các bảng định tuyến tại mỗi nút lưu các thông tin cơ bản như định danh nút đích (*Destination\_id*), định danh nút tiếp theo (*Next\_node*), cổng ra tương ứng (*Output port*).

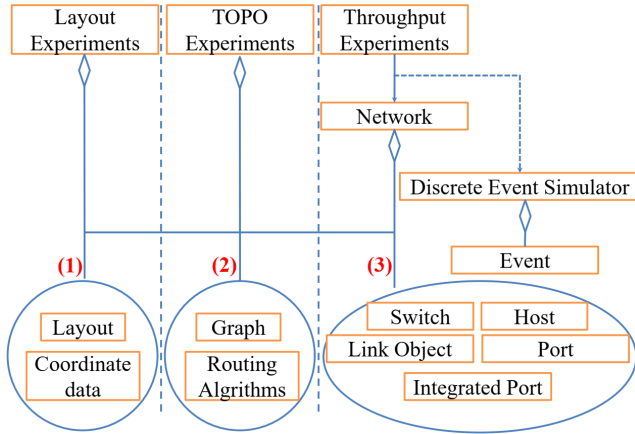
Quyết định định tuyến: thông tin nút đích  $T$  được cập nhật vào phần tiêu đề gói tin (header packet). Tại mỗi nút trung gian  $u = (S, u_1, u_2, \dots, T)$ , một kỹ thuật tra cứu tương thích dài nhất (Longest Prefix Matching Lookup Table) [25] được sử dụng để so sánh định danh nút đích trong phần tiêu đề gói tin với các định danh nút đích được lưu trữ trong bảng định tuyến của nút  $u$ . Quyết định định tuyến được thực thi sau khi xác định nút tiếp theo và cổng ra tương ứng tại nút  $u$ , được mô tả tại hình 5.

### 4) Thiết kế lớp chi tiết:

Mạng thực tế được cấu trúc bởi các thiết bị chính như switch, router, host, wire. Hình 6 mô tả thiết kế chi tiết các thành phần của gói mạng thực tế. Để giả lập được hoạt động của mạng, ta cần trừu tượng hóa các thành phần của mạng (như host, switch, link, packet) như mô tả tại hình 6.



Hình 5. Lưu đồ quyết định định tuyến.



Hình 6. Thiết kế kỹ thuật chi tiết của SSiNET.

- Host: máy chủ trong TTDL, có chức năng gửi và nhận các gói tin.
- Switch: thiết bị liên kết các máy chủ với nhau, cần dựa vào mô-đun định tuyến để tìm đường đi giữa các switch trên mạng.
- Link: đường liên kết giữa các thành phần đầu cuối của mạng (network terminal), mỗi liên kết có băng thông riêng.

### 3. Các vấn đề đang hoàn thiện

Đây là một dự án đang tiếp tục được thực hiện và trong bài báo này chúng tôi trình bày những kết quả đã đạt được: cách tiếp cận, kiến trúc tổng quan hệ thống và một số mô-đun lõi, tập trung vào nhóm các thực nghiệm đánh giá tô-pô trên các hiệu năng đặc tính đồ thị và định tuyến. Chúng tôi đang tiếp tục hoàn thiện phần các thực nghiệm sức tải và thông lượng (traffic related) với trung tâm là các vấn đề về xử lý tương tranh và dòng đợi, và dự định sẽ công bố các

kết quả mới trong một bài báo trong tương lai. Để hoàn thiện bức tranh thiết kế tổng quan, chúng tôi giới thiệu ở đây một số nét ý tưởng chính.

#### 1) Đánh giá hiệu năng làm việc khi có tải:

Với phương châm mô phỏng giảm lược, chúng tôi chỉ quan tâm quản lý các sự kiện rời rạc (discrete event) xung quanh việc phát và thu nhận các gói tin. Mô hình thu phát cho phép xác định thời gian kỳ vọng phát hay thu một gói tin theo độ dài đã biết, từ đó mỗi lời gọi phát hay thu gói tin sẽ làm tạo ra, bởi bộ lập lịch điều khiển sự kiện, một gói tin sự kiện theo cấu trúc cho trước với thời điểm kích hoạt sự kiện tính trước được và được đưa vào dòng xếp hàng. Nhờ có nhãn thời gian (thời điểm cần xử lý nói trên) dòng sự kiện sẽ được xâu chuỗi giúp mô phỏng một quá trình thu phát của một hệ phân tán. Sự hạn chế về băng thông của mỗi liên kết, hay khả năng xử lý tiếp nhận tại mỗi switch sẽ được mô phỏng bởi việc thành lập và quản lý các dòng xếp hàng của các yêu cầu (request) gửi đến. Các dòng xếp hàng là có kích thước cho trước nên khi đầy có thể từ chối nhận thêm yêu cầu. Việc xây dựng các thành phần đối tượng cốt lõi nói trên sẽ đảm bảo việc mô phỏng giảm lược một hệ thống mạng phân tán trong đó các chuỗi xử lý thu phát tin sẽ được xâu chuỗi hợp lý để tạo ra dòng thời gian xử lý sự kiện, phản ánh được thực tiễn một cách hợp lý. Những nghiên cứu tiếp theo dự định sẽ cho phép xây dựng công cụ đảm bảo đánh giá trung thực các tiêu chí hiệu năng liên quan đến tải giao thông như sức tải và thông lượng của MLK.

#### 2) Sử dụng phương pháp xấp xỉ:

Các thực nghiệm đối với tô-pô, đặc biệt là qua mô phỏng, hiện nay mới chỉ được tiến hành giả lập trên một chip tính toán (CPU) với bộ nhớ (RAM) hữu hạn. Do đó, rất khó có khả năng thực hiện với mạng kích thước lớn, ngay cả khi áp dụng phương pháp giảm lược mà chúng tôi đã đề xuất ở trên. Để nâng tính khả thi thực nghiệm đánh giá các tô-pô mạng kích thước lớn hơn nữa, chúng tôi đề xuất bổ sung kỹ thuật tính toán xấp xỉ trong đánh giá các đặc tính đồ thị. Kỹ thuật này khai thác một đặc điểm quan sát thấy là trong các MLK kiểu mới, đặc biệt là các tô-pô được tạo theo mô hình ngẫu nhiên, việc xây dựng định tuyến giữa hai cặp nguồn–đích bất kỳ là hầu như độc lập với nhau. Ví dụ, nhìn chung xây dựng đường đi giữa một cặp nút  $(u, v)$ , không phụ thuộc vào việc xây dựng cho một cặp  $(u', v')$  khác. Từ đó, ta thấy không cần thiết phải có một quá trình khởi đầu xây dựng bảng định tuyến cho tất cả nút mạng, mà có thể tiến hành *on-the-fly*, tức là khi nào có một nhiệm vụ định tuyến nảy sinh thì các bảng định tuyến cho các nút nguồn và trung gian sẽ được xây dựng phục vụ trực tiếp cho nhu cầu tại chỗ (và tích lũy cho sau này). Hơn nữa một số đặc tính hiệu năng như ARPL có thể tính xấp xỉ bằng cách: thay vì tính độ dài đường đi của tất cả các



cặp nút nguồn– đích có thể rồi lấy trung bình, chỉ cần thực hiện với một tỷ lệ tương đối thấp (vài phần trăm) các cặp nguồn–đích. Phần IV dưới đây sẽ báo cáo một số kết quả thực nghiệm khả quan khi chúng tôi khảo sát áp dụng kỹ thuật xấp xỉ này.

**IV. KẾT QUẢ THỰC NGHIỆM**

Đây là một dự án phần mềm có yêu cầu tương đối cao nên chúng tôi mới thực hiện được một số khối chức năng cơ bản, trong đó có khối chức năng đánh giá trên phương diện phân tích đồ thị. Sau đây chúng tôi trình bày một số kết quả thực nghiệm với hệ công cụ về phân tích đồ thị, một trong những yêu cầu đánh giá cốt lõi nhất. Chúng tôi cũng minh họa sự hiệu quả của công cụ mô phỏng SSiNET bằng cách giả lập các mạng với kích thước khác nhau, và so sánh với thời gian thực thi các giả lập đó trên bộ công cụ NS3 [15].

Với mỗi loại tô-pô cụ thể, chúng thường được áp dụng một vài thuật toán định tuyến để đánh giá hiệu năng MLK. Do vậy, trong các mục IV.1 và IV.2, chúng tôi thử nghiệm với các tô-pô và thuật toán định tuyến tương ứng như sau: FatTree [12] – SPR, Kleinberg – Small-world – SPR, 2-D Torus [1] – Greedy, RSN – TZ [18]. Kết quả thực nghiệm không đánh giá cặp tham số tô-pô và thuật toán định tuyến nào hiệu quả hơn, mà chỉ xác định khả năng thực nghiệm của SSiNET. Chúng tôi lần lượt thí nghiệm trên các bộ kích thước  $2^{10}$ ,  $2^{11}$ ,  $2^{12}$ ,  $2^{13}$ . Ngoài ra, do cấu trúc đặc thù của FatTree [12], chúng tôi sử dụng kích thước 2000 và 4394 tương ứng với  $2^{11}$  và  $2^{12}$  tương ứng với các tô-pô còn lại.

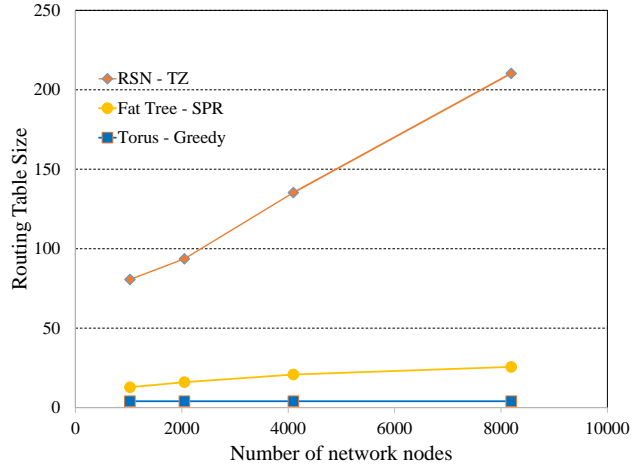
Ở thời điểm hiện nay sản phẩm công cụ của chúng tôi mới tập trung vào lớp thực nghiệm phân tích đồ thị, nên chưa có nhiều thông tin để so sánh đánh giá năng lực thực thi với các công cụ khác, ngoại trừ một so sánh thu được khi đang triển khai bước đầu phần mô phỏng thông lượng tại mục IV.3.

**1. Kích thước bảng định tuyến**

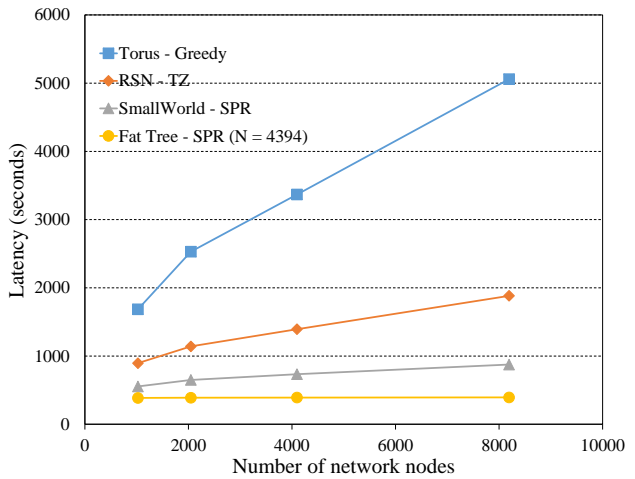
SPR xây dựng bảng định tuyến với kích thước  $O(N)$ , với  $N$  là kích thước mạng. Do đó, trong thử nghiệm, chúng tôi không trình bày kết quả của SPR trên tô-pô Kleinberg – Smallworld. Hình 7 biểu diễn RTS của 2-D Torus không đổi với mọi kích thước mạng, do tính chất cấu trúc vòng đặc thù của Torus. Trong khi đó, FatTree – SPR có RTS đạt 1,25% và 0,31% kích thước mạng  $2^{10}$  và  $2^{13}$  tương ứng, RSN – TZ[18] có RTS 7,78% và 2,57% tương ứng.

**2. Độ trễ truyền tin (Latency)**

Chúng tôi tiến hành thực nghiệm tính toán độ trễ với các tham số độ trễ trên switch là 100 nano giây, độ trễ



Hình 7. Tính toán kích thước bảng định tuyến.



Hình 8. Độ trễ truyền tin trên mạng.

trên đường truyền cáp quang là 5 nano giây trên mỗi mét dài [1, 13]. Độ trễ truyền tin toàn mạng  $L$  được tính toán dựa trên công thức sau:

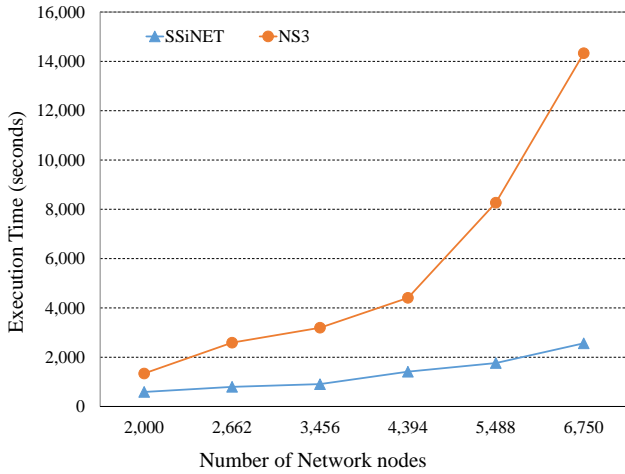
$$L = \sum_{i=1}^N \sum_{j=1}^N (Path_{hop}(i, j) * 100 + Path_{length}(i, j) * 5), \quad (1)$$

trong đó,  $Path_{hop}(i, j)$  là số “hop”<sup>4</sup> trên đường đi ngắn nhất và  $Path_{length}(i, j)$  là chiều dài đường đi ngắn nhất giữa  $i$  và  $j$ . Với tổng số cặp đường đi  $\Sigma pair(i, j)$ , độ trễ trung bình được tính theo công thức:

$$AvgLatency = \frac{L}{\Sigma pair(i, j)}. \quad (2)$$

Hình 8 chỉ ra kết quả thử nghiệm tính toán độ trễ. Trong đó, với cấu trúc đặc thù dạng cây, đường kính mạng bé, FatTree [12], đạt độ trễ thấp nhất trong mọi kích thước

<sup>4</sup>Số “khoảng” giữa hai nút liền kề trên đường định tuyến.



Hình 9. Thời gian thực thi của SSiNET và NS3.

Bảng I  
ĐƯỜNG KÍNH MẠNG VỚI CÁC TỈ LỆ KÍCH THƯỚC MẠNG

Diameter	1024	2048	4096	8192
1%	6,30	7,00	7,90	8,00
2%	6,80	7,00	8,00	8,00
5%	7,00	7,00	8,00	8,00
10%	7,00	7,00	8,00	8,00
100%	7,00	7,00	7,90	8,00

thử nghiệm. Trong khi đó, Torus có độ trễ cao do sử dụng định tuyến (DOR: Dimension Order Routing [1]), số hop trên đường định tuyến cao hơn so với các mạng ngẫu nhiên Kleinberg – Smallworld.

Các kết quả so sánh tương đối giữa các tô-pô (và thuật toán định tuyến) nói trên phản ánh khớp với nhận định của chúng tôi thu được từ những khảo sát thông qua [12, 18, 20].

### 3. Kết quả bước đầu về đánh giá thông lượng

Cuối cùng, chúng tôi minh họa việc giả lập của MLK Fat Tree [12] với việc giá trị  $k$  thay đổi từ 20 đến 30. Để đo đặc hiệu năng của SSiNET, chúng tôi sử dụng máy tính Intel Core i7 2,3 GHz với bộ nhớ chính 8 GB. Hình 9 cho thấy thời gian thực thi của chương trình giả lập hoạt động trên SSiNET và NS3 [15], và hình này cung cấp cho người đọc thấy rõ ràng rằng SSiNET chỉ mất thời gian khoảng 40 phút (2564 giây) để giả lập mạng Fat-tree với  $k = 30$  (ở đây mạng sẽ có  $k * k * k * k / 4 = 6750$  nodes) trong khi NS3 yêu cầu tới khoảng 4 giờ.

### 4. Phương pháp xấp xỉ

Trong phương pháp này, chúng tôi tiến hành lấy ngẫu nhiên một số cặp nguồn–đích theo một tỉ lệ  $p\%$  so với

Bảng II  
ĐỘ TRỄ TRUNG BÌNH

AvgLatency	1024	2048	4096	8192
1%	536,80	628,24	712,59	850,08
2%	536,05	628,11	712,35	850,46
5%	535,92	628,13	712,40	850,48
10%	536,15	628,22	712,31	850,49
100%	612,85	767,49	723,18	850,41

Bảng III  
THỜI GIAN THỰC THI

Kích thước mạng	1024		8192	
Tỉ lệ thử nghiệm	100%	5%	100%	5%
Diameter	7,00	7,00	8,10	8,00
ARPL	4,51	4,52	5,93	5,93
AvgLatency	612,85	535,92	850,08	850,48
RunTime	5,74	1,02	1083,38	68,24

kích thước mạng. Tập các ứng viên được tính toán theo công thức  $(N * (N - 1)) * p\%$ ,  $p=1, 2, 5, 10$ , với  $N$  là kích thước mạng.

Bảng I cho thấy kết quả thực nghiệm tính đường kính mạng với các tỉ lệ 1%, 2%, 5%, 10%, 100%. Trường hợp 100% là tính toán toàn bộ các cặp nguồn đích, tức là thực hiện đầy đủ như thông lệ. Các trường hợp 1% và 2% có giá trị đường kính mạng sai khác 10% và 3% với trường hợp 100%, tương ứng. Tuy nhiên với tỉ lệ 5% và 10% thì sự sai khác không có hoặc tương đối nhỏ so với 100%. Ngay cả độ trễ trung bình cũng chỉ sai khác trên 10%, đặc biệt với mạng kích thước 4096, sự sai khác đó chỉ là 1% và 2% tương ứng như Bảng II.

Với phương pháp xấp xỉ như trên, thời gian thực thi của mạng giảm đáng kể. Ví dụ, thời gian thực thi giảm 82,25% và 93,70% với kích thước mạng 1024 và 8192 tương ứng, như Bảng III. Những kết quả thực nghiệm này cho thấy rõ phương pháp thực nghiệm xấp xỉ đã đề xuất là hứa hẹn và nên được triển khai đầy đủ.

## V. KẾT LUẬN

Trong nghiên cứu này, chúng tôi đã đề xuất một cách tiếp cận mới cho việc thiết kế công cụ thực nghiệm đánh giá các tô-pô MLK theo phương châm mô phỏng giảm lược và trình bày một bản thiết kế tổng quan và chi tiết cho sản phẩm công cụ SSiNET, phát triển theo tiếp cận này. Công cụ cho phép đánh giá và so sánh các tô-pô có kích thước lớn trên các phương diện truyền thống như các đặc tính đồ thị và định tuyến và đặc tính khai thác có tải. Ngoài ra khi đã hoàn thiện, công cụ còn cho phép tiến hành thực nghiệm

đánh giá bố trí triển khai trên mặt sàn. Qua đó cho phép các nhà nghiên cứu thử nghiệm các thiết kế tô-pô mới một cách đa dạng và linh hoạt mà không phải phát triển chương trình riêng để cài đặt các tô-pô hoặc thuật toán định tuyến có sẵn.

Sản phẩm mới chỉ được thực hiện ở phần cơ bản, nên trong bài báo này chúng tôi tập trung giới thiệu thiết kế chi tiết các phần cơ bản đó và giới thiệu một vài kết quả thực nghiệm liên quan. Trong giai đoạn phát triển tiếp theo, chúng tôi sẽ tiếp tục hoàn thiện SSINET, trong đó có việc hoàn thiện các chức năng mô phỏng cho phép đánh giá các yếu tố hiệu năng quan trọng khác như thông lượng mạng. Đồng thời kỹ thuật và phương pháp thực nghiệm xấp xỉ sẽ được hoàn thiện hơn để nâng cao hơn nữa tính khả thi của công cụ với việc đánh giá MLK có kích thước lớn và rất lớn.

## GHI NHẬN TÀI TRỢ

Nghiên cứu này được tài trợ bởi Trường Đại học Bách khoa Hà Nội trong đề tài mã số T2017-LN-15.

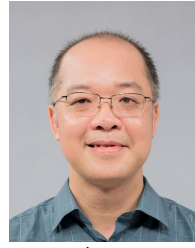
## TÀI LIỆU THAM KHẢO

- [1] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.
- [2] M. Koibuchi, H. Matsutani, H. Amano, D. F. Hsu, and H. Casanova, "A case for random shortcut topologies for HPC interconnects," in *2012 39th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2012, pp. 177–188.
- [3] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 339–350, 2011.
- [4] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.
- [5] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 75–86.
- [6] L. Gyarmati and T. A. Trinh, "Scafida: A scale-free network inspired data center architecture," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 5, pp. 4–12, 2010.
- [7] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang, "MDCube: a high performance network structure for modular data center interconnection," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 25–36.
- [8] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," in *ACM SIGCOMM computer communication review*, vol. 39, no. 4. ACM, 2009, pp. 51–62.
- [9] J.-Y. Shin, B. Wong, and E. G. Sirer, "Small-world data centers," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*. ACM, 2011, p. 2.
- [10] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, 2012, pp. 225–238.
- [11] Y. Yu and C. Qian, "Space shuffle: A scalable, flexible, and high-bandwidth data center network," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 13–24.
- [12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 63–74.
- [13] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova, "Skywalk: A topology for HPC networks with low-delay switches," in *2014 IEEE 28th International Parallel and Distributed Processing Symposium*. IEEE, 2014, pp. 263–272.
- [14] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *2008 International Symposium on Computer Architecture*. IEEE, 2008, pp. 77–88.
- [15] D. Wong, K. T. Seow, C. H. Foh, and R. Kanagavelu, "Towards reproducible performance studies of datacenter network architectures using an open-source simulation approach," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2013, pp. 1373–1378.
- [16] "OMNeT++ discrete event simulator." [Online]. Available: <https://www.omnetpp.org>
- [17] "Simgrid: Versatile simulation of distributed systems." [Online]. Available: <http://simgrid.gforge.inria.fr>
- [18] M. Thorup and U. Zwick, "Compact routing schemes," in *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 2001, pp. 1–10.
- [19] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large-scale sensor networks with complex shape," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, p. 31, 2009.
- [20] M. Benito, E. Vallejo, and R. Beivide, "On the use of commodity ethernet technology in exascale HPC systems," in *2015 IEEE 22nd International Conference on High Performance Computing (HiPC)*. IEEE, 2015, pp. 254–263.
- [21] L. J. Cowen, "Compact routing with minimum stretch," *Journal of Algorithms*, vol. 38, no. 1, pp. 170–183, 2001.
- [22] "Network simulation: NS2." [Online]. Available: <https://ns2tutor.weebly.com>
- [23] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE journal of solid-state circuits*, vol. 41, no. 3, pp. 712–727, 2006.
- [24] H. Liu, "Routing table compaction in ternary CAM," *IEEE Micro*, vol. 22, no. 1, pp. 58–64, 2002.
- [25] C. Basso, J. L. Calvignac, G. T. Davis, and P. C. Patel, "Longest prefix match lookup using hash function," Apr. 2010, US Patent 7,702,630.



**Kiêu Thành Chung** tốt nghiệp Kỹ sư Công nghệ Thông tin năm 2003 và Thạc sĩ Công nghệ Thông tin năm 2010 tại Trường Đại học Bách Khoa Hà Nội. Năm 2016, tác giả nghiên cứu tại Viện Công nghệ Thông tin Quốc gia Nhật Bản (NII), hiện nay là nghiên cứu sinh tại Viện Công nghệ Thông tin và Truyền thông (SoICT), Trường Đại

học Bách khoa Hà Nội. Lĩnh vực nghiên cứu bao gồm: mạng liên kết, thuật toán định tuyến.



**Nguyễn Khanh Văn** tốt nghiệp Kỹ sư Tin học tại Trường Đại học Bách Khoa vào năm 1992, Thạc sĩ Khoa học Máy tính tại Đại học Wollongong (Úc) vào năm 2000 và Tiến sĩ Khoa học Máy tính tại Đại học California-Davis (Mỹ) vào năm 2006. Hiện nay ông là Phó Giáo sư, giảng dạy và nghiên cứu tại Viện Công nghệ Thông tin

và Truyền thông, Trường Đại học Bách khoa Hà Nội. Các lĩnh vực nghiên cứu chính bao gồm: thuật toán và các mô hình lý thuyết trong tính toán phân tán và mạng máy tính (mạng liên kết, mạng cảm biến không dây), an toàn thông tin.



**Nguyễn Tiến Thành** tốt nghiệp Kỹ sư Công nghệ Thông tin năm 2008 và Thạc Sĩ ngành Công nghệ Thông tin năm 2011 tại Đại học Bách Khoa Hà Nội. Hiện nay tác giả đang là giảng viên tại Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách khoa Hà Nội. Lĩnh vực nghiên cứu của tác giả bao gồm: công nghệ phần

mềm, kiểm chứng mô hình và mạng tự điều khiển (self-driving networks).