Xây dựng tô-pô mạng liên kết 3 chiều dựa trên kiến trúc DSN nhằm thích ứng cài đặt thực tế

Constructing 3D Interconnection Network Topology Using The DSN Framework for Achieving Layout-Awareness

Lê Xuân Thống Nhất, Nguyễn Khanh Văn

Abstract: *The highly rising demands today in high performance computing or building big data centers make most of traditional interconnection network topologies become outdated. Currently, to develop new types of high performance network, random topologies such as using random shortcuts, small-world models, are promising research directions.*

In this paper, we develop a new 3D topology named 3D-DSN based on the principles of Distributed Shortcut Network (DSN) [1] to support saving cable length. The main idea is to use a 2D-Torus as the base structure instead of a 1D-Ring as in [1]. Furthermore, the set of shortcuts are distributed in both directions of 2D-Torus and the routing logic is extended to 5 phases instead of 3 phases as in [1]. We also give the analyses of topology properties in both theoretical and experiment aspects, then conclude that 3D-DSN is more realistic than the basic DSN and provide a nice trade-off between network diameter (or average path length) and cable cost.

Keyword: *Interconnection networks, custom routing, layout-awareness, distributed shortcut*

I. GIỚI THIỆU

Nghiên cứu tô-pô của các mạng liên kết là một chủ đề cơ bản truyền thống trong lĩnh vực kiến trúc mạng máy tính, tính toán song song và tính toán hiệu năng cao. Với sự phát triển nhanh chóng hiện nay trong các lĩnh vực này, hầu hết các cấu *trúc tô-pô truyền thống đã trở nên lạc hậu*. Chẳng hạn đa phần các tô-pô truyền thống không có một cân bằng tối ưu giữa đường kính đồ thị và số liên kết tại mỗi nút. Vì thế khi mạng có số nút tính toán lớn, để khống chế số liên kết đủ nhỏ tại mỗi đỉnh (do yêu cầu kỹ thuật của các switch), đường kính đồ thị cũng như độ dài trung bình đường đi ngắn nhất sẽ không nhỏ, dẫn đến *độ trễ truyền tin* là đáng quan ngại, làm giảm khả năng xử lý của cả hệ thống.

Mặt khác, các tô-pô truyền thống tiệm cận tối ưu các yêu cầu hiệu năng nói trên lại đòi hỏi thiết kế phức tạp và cứng nhắc với số nút chọn trước theo công thức (ví dụ như tô-pô *k-ary n-cube* chỉ cho phép số nút k^n), điều này làm ảnh hưởng rất nhiều đến việc mở rộng dần mạng lưới nút cũng như thay đổi bảo trì, điều đặc biệt quan trọng với các trung tâm dữ liệu kiểu mới. Vì vậy, gần đây nhiều nghiên cứu mới đang được đẩy mạnh để tìm ra những tô-pô mạng liên kết kiểu mới đáp ứng tốt hơn cho những ứng dụng hiện đại.

Trong lĩnh vực nghiên cứu cơ bản này, một hướng nghiên cứu mới được quan tâm gần đây là ứng dụng các mô hình đồ thị ngẫu nhiên (random graphs), trong đó có các mô hình thế-giới-nhỏ (small-worlds), để tạo ra các tô-pô kiểu mới với nhiều tính năng ưu việt hơn. Các mô hình đồ thị hiện đại này có 2 ưu điểm cơ bản như sau. Thứ nhất, chúng hướng tới các hiệu năng ưu việt như đường kính nhỏ, số liên kết tại đỉnh nhỏ và những thuật toán tìm đường đơn giản mà hiệu quả. Thứ hai, tính ngẫu nhiên của việc tạo liên kết (trên cơ sở một phân bố xác suất xác định cho trước) tạo nên

tính mềm dẻo, thuận lợi để xây dựng các tô pô liên kết với khả năng co giãn qui mô cao (scalability).

Bài báo này tiếp tục khai phá hướng đi nói trên. Dựa trên mô hình Distributed Shortcut Networks (DSN) được đề xuất trong [1], chúng tôi xây dựng một tô-pô mạng liên kết mở rộng với đặc thù khai thác không gian 3 chiều, tạo lợi thế phù hợp với việc bố trí cài đặt trong thực tế. Ý tưởng mở rộng của chúng tôi là lấy một 2D-Torus làm cơ sở thay vì 1D-Ring như trong [1]. Việc xây dựng các liên kết xa vì thế được tiến hành theo 2 phương, đồng thời thuật toán định tuyến được phát triển theo 5 pha thay vì 3 pha như trong [1]. Kết quả khảo sát và đánh giá bước đầu cho thấy 3D-DSN là gần gũi với ứng dụng thực tế hơn so với DSN cơ bản, cho phép cân bằng hiệu quả giữa đường kính đồ thị (và độ dài trung bình đường đi ngắn nhất) với chi phí cáp mạng.

Cấu trúc của bài báo như sau. Trong Phần II chúng tôi sẽ trình bày tổng quan về lĩnh vực, và sau đó trong Phần III sẽ tóm tắt những nét cơ bản của mô hình kiến trúc DSN nói trên. Phần IV trình bày chi tiết về đề xuất mới của chúng tôi cùng phần khảo sát các tính chất tô-pô trên lý thuyết. Phần V trình bày các kết quả khảo sát thực nghiệm ban đầu. Phần VI cung cấp kết luận chung. Một số chứng minh lý thuyết các tính chất của tô-pô mới được đưa vào phụ lục tham khảo.

Trong bài báo, chúng tôi cố gắng lựa chọn sử dụng một số thuật ngữ tiếng Anh khá thông dụng trong lĩnh vực chuyên môn vì khó dịch sát nghĩa và hay. Đó là các thuật ngữ: *Shortcut* (nối tắt); *Ring* (vòng); *Torus* (lưới vòng); Cabinet (tủ chứa nút/lõi tính toán); Node (nút tính toán); *Super-node* (siêu nút); *Switch* (chuyển mạch). Các cặp thuật ngữ Anh-Việt như Node và nút, hay Super-node và siêu nút được dùng song song, thay thế lẫn nhau, tùy thuộc vào tình huống câu văn mô tả.

II. TỔNG QUAN LĨNH VỰC

Từ lâu các nhà khoa học đã đặc biệt quan tâm đến vấn đề thiết kế tô-pô dùng bậc đỉnh thấp (low-radix) hay bậc đỉnh cao (high-radix) cho các hệ thống tính toán rất lớn. IBM đã phát triển các 3D torus cho siêu máy tính BlueGene/L và 5D torus cho BlueGene/Q với các mạng bậc thấp, và Dragonfly [4] cho Power 775 với mạng bậc cao. Lịch sử phát triển cho thấy các tô-pô bậc thấp đã được sử dụng phổ biến trong các hệ thống tính toán hiệu năng cao nhờ có các ưu điểm như: (i) cơ chế quản lý lỗi đơn giản, (ii) dễ dàng tích hợp định tuyến mạng và các lõi tính toán vào từng chip xử lý hay mạch đơn, (iii) cách bố trí switch mạng trong không gian vật lý tương đối tiết kiệm và giao thức truyền thông dễ chỉnh sửa. Chính vì vậy mà 6 trên 10 các siêu máy tính trong xếp hạng TOP500 vào tháng 6/2012 sử dụng các phiên bản của torus. [8]

Để đạt được chỉ số tốt về đường kính mạng và bậc của đỉnh, các thiết kế tô-pô bậc thấp truyền thống thường sử dụng kiến trúc phân cấp hoặc hoán vị đỉnh đồ thị, ví dụ các tô-pô dựa trên phép bố trí lại các đỉnh như đồ thị De Bruijn [17], các đồ thị (n, k)-star [18]. Đối với các cấu trúc phân tầng, những nghiên cứu trước đây cũng hướng tới chỉ số bậc của đỉnh thấp. Các tô-pô bậc thấp điển hình bao gồm Hypernet [19] được tạo thành từ việc kết nối các mạng con theo đồ thị đầy đủ, các Cube Connected Cycles (CCC) được xây dựng bằng cách thay thế mỗi đỉnh trong hypercube bằng 1 vòng tròn (ring) các đỉnh. Mặc dù các mạng này có ưu điểm hấp dẫn về chỉ số đường kính – bậc, việc cài đặt thực tế trong không gian máy tính vật lý là không hề đơn giản, có thể dẫn đến chi phí cài đặt và vận hành không nhỏ.

Trong các nghiên cứu trước đây, các đồ thị ngẫu nhiên đã được biết đến với ưu điểm về đường kính nhỏ. Đặc tính "thế giới nhỏ" của những mạng này đã nhận được nhiều sự quan tâm trong giới khoa học, bắt nguồn từ bài báo kinh điển của Watts và Strogatz [20]. Hiệu ứng thế giới nhỏ nhìn chung thể hiện một mạng có đường kính nhỏ, bậc thấp và hỗ trợ tìm kiếm phân tán, nghĩa là tìm đường chỉ sử dụng thông tin cục bộ. Kleinberg trong bài báo [15] về hiệu ứng thế giới nhỏ từ góc nhìn giải thuật đã chỉ ra rằng việc tăng cường có lựa chọn một số liên kết đi xa có thể giúp giảm thiểu nhanh chóng đường kính mạng và cung cấp khả năng tìm kiếm phân tán. Chính vì vậy, đồ thị thế giới

nhỏ và ngẫu nhiên đã được đề xuất để xây dựng các trung tâm dữ liệu với khả năng mở rộng, kháng lỗi và thông lượng cao trong các nghiên cứu gần đây [9, 10].

Trong số những tô-pô ảnh hưởng từ hiệu ứng này, mạng Distributed Loop Network (DLN-*x*) với bậc *x* [2], gồm có n đỉnh được sắp xếp trên 1 đường tròn và được thêm các đường nối tắt giữa các đỉnh *i* và *j* mà $j = i + \lfloor n/2^k \rfloor \; mod \; n$ với $k = 1, \ldots, x - 2$. Đồ thị này có đường kính là logarithmic với $x = logn$. Koibuchi và các đồng nghiệp [2] cũng đã giới thiệu tô-pô gọi là DLN-*x*-*y* trong đó *y* là số liên kết ngẫu nhiên được thêm vào mỗi node trong DLN-*x*. Những đồ thị này có bậc hằng số và đường kính logarithmic với *x*, *y* cố định. Mạng thế giới nhỏ của Kleinberg trong [15] lại gồm 1 lưới ô vuông với các đường nối tắt được thêm vào cũng đã thể hiện đầy đủ ưu điểm của hiệu ứng thế giới nhỏ. Tuy nhiên, những tô-pô này có những nhược điểm quan trọng, DLN-*logn* có đường kính logarithmic nhưng bậc *logn* lớn, DLN-*x*-*y* có đường kính logarithmic và bậc hằng số nhưng giải thuật tìm đường yêu cầu những thông tin của toàn bộ tô-pô, và quan trọng nhất là chi phí cáp nối là quá lớn. Trong khi đó, mặc dù có số lượng cáp nối nhỏ hơn, bậc hằng số, đường kính logarithmic nhưng mạng thế giới nhỏ của Kleiberg sử dụng tìm kiếm tham lam với độ dài đường đi chỉ là bình phương của kết quả tối ưu [16].

Để giảm thiểu chi phí cáp nối trong mạng, có 2 phương pháp đã được đề xuất bởi Koibuchi và đồng sự [7], [11], trong đó người ta tạo ra các tô-pô dựa ngẫu nhiên có độ dài đường đi nhỏ gần như các tô-pô ngẫu nhiên hoàn toàn và cải thiện cách bố trí của chúng trong không gian vật lý. Một trong 2 phương pháp sử dụng ý tưởng ngẫu nhiên hóa các liên kết mạng sau khi tối ưu bố cục vật lý, trong khi phương pháp kia tối ưu bố cục vật lý sau khi đã ngẫu nhiên hóa các liên kết mạng. Tuy nhiên, theo quan điểm của chúng tôi, việc sử dụng một phân bố ngẫu nhiên tổng thể không chú ý đến yếu tố địa lý không gian có thể dẫn đến việc sử dụng những đường liên kết quá dài, không cần thiết.

Trong [1] các tác giả đã đề xuất mô hình mạng liên kết Distributed Shortcut Network (DSN) với ý tưởng khai thác hiệu ứng thế giới nhỏ dù vẫn xây dựng một tô-pô với các liên kết cố định. Dựa trên một một đồ thị ring ban đầu, ý tưởng này đề xuất việc đưa vào các liên kết xa, tuyển chọn theo cách thức đặc biệt để tạo ra hiệu ứng thế-giới-nhỏ, trong khi vẫn đáp ứng yêu cầu bậc hằng số. Tuy nhiên DSN còn có nhược điểm: việc bố trí các nút theo một vòng tròn vật lý là không thực tế và nếu miễn cưỡng cài đặt như một vòng tròn logic thì cũng gây tốn kém cáp mạng không cần thiết. Bài báo của chúng tôi sẽ khắc phục vấn đề này bằng việc xem xét các nút như các đối tượng trong không gian 3 chiều, hay nói đúng hơn là xuất phát từ một lưới 2 chiều của các cabinet – chiều thứ 3 còn lại là chiều xếp đứng của các nút tính toán trong một cabinet. Việc thiết kế các liên kết xa sẽ được áp dụng theo hai phương thay vì một phương như trước kia. Nhờ đó tô-pô đề xuất sẽ đạt hiệu quả tốt hơn, đặc biệt là giảm chi phí cáp mạng.

III. LƯỢC ĐỒ THIẾT KẾ DISTRIBUTED SHORTCUT NETWORK – DSN

Chúng tôi xin trình bày tóm tắt về mô hình DSN mà chúng tôi dựa theo để phát triển đề xuất mới này.

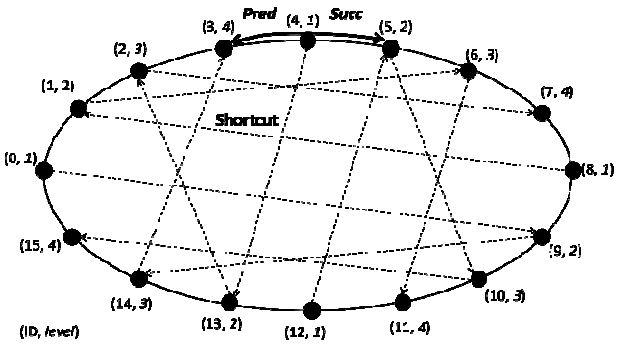
DSN được cấu tạo như một *ring* gồm *n* nút được nhóm thành các siêu nút liên tiếp kích thước $p = log\, n$. Mỗi nút có 2 liên kết ngắn vô hướng với 2 nút trước và sau trên ring và có thể có một liên kết xa đi ra được gọi là *shortcut*. Mỗi siêu nút có tối đa *p*-1 *Shortcut* đi ra với độ dài tương ứng $n/2^k$ với $k = 1 \ldots log\, n$ (ở đây độ dài tính theo số nút đi qua trên *ring*). Trong quá trình định tuyến, từ tập các *Shortcut* này người ta chọn ra *Shortcut* lân cận có khả năng đi xa một khoảng bằng xấp xỉ nửa đường đi tới đích.

Dưới đây, chúng tôi sẽ trình bày tóm tắt cấu trúc của tô-pô DSN-*x*-*n* với *n* nút, hay ngắn gọn là DSN-*x*, mà ở đó tham số *x* được chọn nằm giữa 1 và *p–1* thể hiện cho tập các *Shortcut* độ dài khác nhau của một super-node:

- **Cấu tạo ring**: *n* nút được sắp xếp trên 1 vòng và mỗi node có một ID từ 0 tới *n*–1. Mỗi node *i* có 2 liên kết vô hướng nội bộ tới node lân cận *(i–1) mod n* và *(i+1) mod n*. Hai liên kết này được lần lượt gọi là *Pred* và *Succ*.
- **Phân loại node:** mỗi nút có một *cấp* (*level*) nằm trong khoảng từ *1* tới *p*. *Cấp* được gán theo một chu kỳ: cấp $i = \overline{1 \dots p}$ ứng với các nút có ID=$k \times p + i - 1$ trong đó $k = 0, 1, 2, \dots, \lfloor n/p \rfloor$. Với mỗi nút cấp *l*, ta đồng thời định nghĩa chiều cao (height) của nút đó bằng *p+1–l*. Theo đó, nút càng cao thì *Shortcut* đi càng xa và nút với chiều cao *p* (cấp 1) có *Shortcut* đi xa nhất, hơn một nửa ring.
- **Shortcut:** Mỗi nút có cấp $l \leq x$ có một liên kết *Shortcut* đi tới nút có cấp *l+1* và có chiều dài theo chiều kim đồng hồ ít nhất $\lfloor n/2^l \rfloor$. Vì vậy ta còn gọi đây là *Shortcut cấp l*.

Để minh họa, Hình 1 mô tả chi tiết cấu trúc mạng DSN-3-16 với 16 nút. Các nút và liên kết ngắn (*Pred*, *Succ*) nằm trên vòng tròn màu xanh. Các liên kết *Shortcut* được minh họa bằng các cạnh màu vàng là các liên kết có hướng. Các nút được gắn nhãn theo cặp (ID, cấp).

Giải thuật định tuyến của DSN được phát triển từ cách thức định tuyến đơn giản giữa các siêu nút của nó (tương tự như DLN-*x* [2] của các siêu nút). Dưới đây chúng ta sẽ trình bày sơ lược ý tưởng chính của giải thuật định tuyến cho DSN, chi tiết có thể tham khảo ở [1]. Giải thuật định tuyến của DSN gồm 3 pha: (**i**) PRE-WORK cho phép nút nguồn *s* tìm kiếm một nút ở phạm vi lân cận *v* với chiều cao phù hợp (đây là pha chuẩn bị cho pha kế tiếp); (**ii**) MAIN-PROCESS liên tục sử dụng các *Shortcut* có khả năng chia đôi đường đi tới đích *t*, nhằm tiến tới phạm vi ngay sát *t;* (**iii**) FINISH di chuyển tới đích sử dụng liên kết nội bộ của siêu nút. Chú ý rằng MAIN-PROCESS là một quá trình lặp: đi xuống liên tục từ node hiện tại *u* để tìm node *v* có chiều cao cần thiết thấp hơn và chọn *Shortcut* chia đôi khoảng cách, sau đó lặp liên tục quá trình này với nút *u* mới. Quá trình lặp chỉ dừng khi không còn *Shortcut* nào khả dụng, nghĩa là nút *u* đã đủ gần đích *t* (thường là đi quá một chút); khi đó ta có thể đơn giản di chuyển bằng *Succ*/*Pred* thẳng tới *t*.

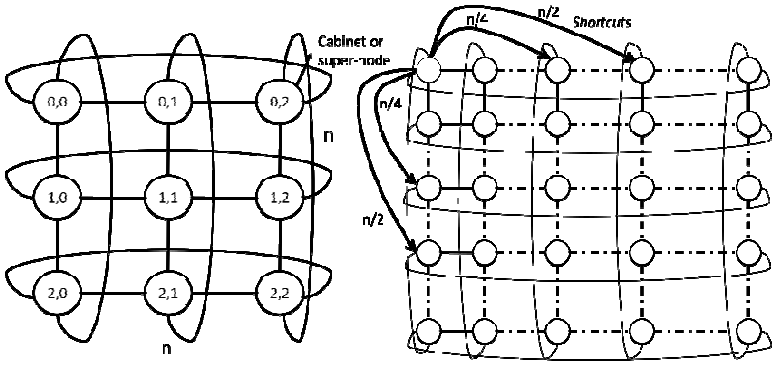


Hình 1. Cấu trúc đầy đủ của DSN-3-16

IV. ĐỀ XUẤT MỚI: TÔ-PÔ 3D-DSN DỰA TRÊN LƯỚI CÁC SIÊU NÚT HAI CHIỀU

DSN là một giải pháp lai giữa tô-pô mạng điển hình (mesh, torus, …) và tô-pô mạng ngẫu nhiên nhằm hướng tới một tô-pô đơn giản nhưng tận dụng được những shortcut đã được tối ưu. Mặc dù vậy, DSN được thiết kế dựa trên một *ring* một chiều cho nên khi triển khai thực tế sẽ cần sắp xếp lại các node, các liên kết cho phù hợp với điều kiện hạ tầng thực tiễn là không gian 3D. Điều này dẫn tới yêu cầu tính toán lại các chỉ số về chiều dài dây, cách bố trí trong một cabinet (tủ chứa các nút xử lý), ... Những khác biệt này là khó khăn không hề nhỏ cho người triển khai, có thể ảnh hưởng lớn tới hiệu năng thực tế.

Ở đây chúng tôi đề xuất 3D-DSN mở rộng từ DSN sao cho thừa kế đầy đủ các điểm mạnh đồng thời tương thích hoàn toàn với môi trường thực tế 3D trong phòng máy, cũng như thiết kế của các cabinet hiện tại. Thay vì dựa trên kiến trúc vòng cơ sở, 3D-DSN được xây dựng theo đúng cách bố trí vật lý 3D của phòng máy mà ở đó các cabinet được sắp xếp theo lưới 2 chiều ngang và dọc còn bản thân chiều đứng là chiều của các nút cùng nằm trong 1 cabinet. Cách xây dựng 3D-DSN sẽ được lần lượt trình bày ở Phần IV.1 trước khi chúng ta đi vào phân tích các đặc điểm của nó ở Phần IV.2.



Hình 2. a-Tô pô cơ sở của các cabinet, b-Bố trí Shortcut giữa các cabinet

IV.1 Thiết kế tô-pô

IV.1.1. Tô-pô cơ sở

Nhìn ở góc độ khái quát, 3D-DSN là tô-pô của các cabinet (gọi là tô-pô cơ sở) mà trong đó mỗi cabinet chứa một số các nút tính toán thông thường. Các cabinet được kết nối dựa theo một torus hai chiều với *n* cabinets theo mỗi chiều ngang hay dọc (Hình 2-a) và ID của các cabinet sẽ được đánh số là (X, Y) dựa trên 2 chiều này. Mỗi cabinet sẽ được thiết kế như là 1 super-node ở trong DSN (2 thuật ngữ này, cabinet và super-node tức siêu nút, sẽ được sử dụng song song với nghĩa tương đương, trừ phi có chú thích rõ).

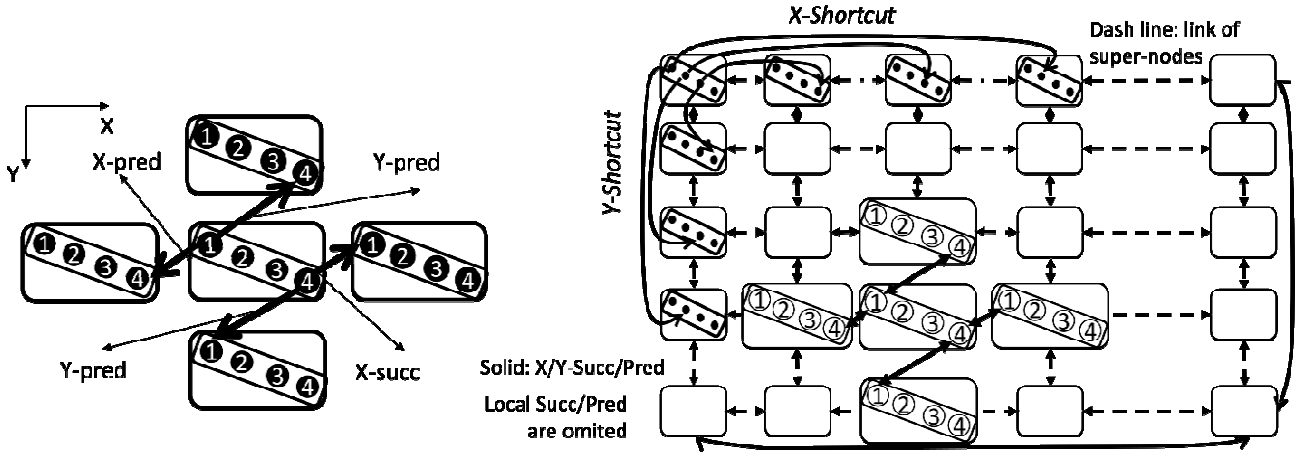
Mỗi cabinet sẽ có *2logn Shortcuts* được chia đều cho 2 chiều X, Y. Ở mỗi chiều, các S*hortcut* lần lượt đi tới các cabinet khác ở khoảng cách $n/2^i$ so với cabinet gốc, với $i = 1 \ldots logn$ (Hình 2-b).

IV.1.2. Tô-pô đầy đủ

Chi tiết hóa thiết kế cơ sở trên, mỗi cabinet là một super-node bao gồm $p = logn$ nút thông thường được kết nối với nhau bởi một ring, nghĩa là mỗi node thông thường có 2 liên kết *pred/succ* tới node liền trước và liền sau trong cùng cabinet (còn gọi là liên kết *intra-cabinet*). Tổng cộng có n^2 cabinet/super-node, mỗi super-node có *logn* nút thông thường nên kích thước mạng là $N = n^2 logn$**.** Dựa trên ID của cabinet, ID của một node mạng thông thường là *(X, Y, i)* trong đó *(X, Y)* là ID của cabinet, *i* là cấp của node bên trong cabinet.

Như đã nói ở phần trước, các cabinet được liên kết với nhau bởi một Torus 2 chiều kết hợp với hệ thống *Shortcut*. Thực chất các liên kết giữa 2 cabinet/super-node ở đây là những kết nối giữa các node thông thường của 2 cabinet (gọi là liên kết *inter-cabinet*). Kết nối thứ nhất nhằm liên kết hai super-node liên tiếp trên torus 2 chiều cơ sở. Chúng được hiện thực hóa nhờ liên kết từ node cuối cùng của một super-node nào đó tới nút đầu tiên của super-node liền sau (gọi là liên kết *X/Y-succ*) và từ nút đầu tiên của super-node nào đó tới nút cuối cùng của super-node liền trước (gọi là liên kết *X/Y-pred*). Mô tả cụ thể các liên kết này như sau (Hình 3-a):

- Mỗi nút thông thường cấp *1* có một *X-pred* và một *Y-pred*
  - *X-pred*: từ *(X,* Y, *1)* tới *(X–1,* Y, *p)* – tới nút có cấp cao nhất của cabinet liền trước theo chiều X.
  - *Y-pred*: từ *(*X, *Y, 1)* tới *(*X, *Y–1, p)* – tới nút có cấp cao nhất của cabinet liền trước theo chiều Y.
- Mỗi node thông thường ở cấp *p* có một *X-succ* và một *Y-succ*
  - *X-succ*: từ *(X,* Y, *p)* tới *(X + 1,* Y, *1)* – tới nút có cấp thấp nhất của cabinet liền sau theo chiều X.
  - *Y-succ*: *(*X, *Y, p)* to *(*X, *Y + 1, 1)* – tới node có cấp thấp nhất của cabinet liền sau theo chiều Y.



Hình 3. a-Kết nối 2 cabinet liền kề. b-Các kết nối liên cabinet (inter-cabinet)

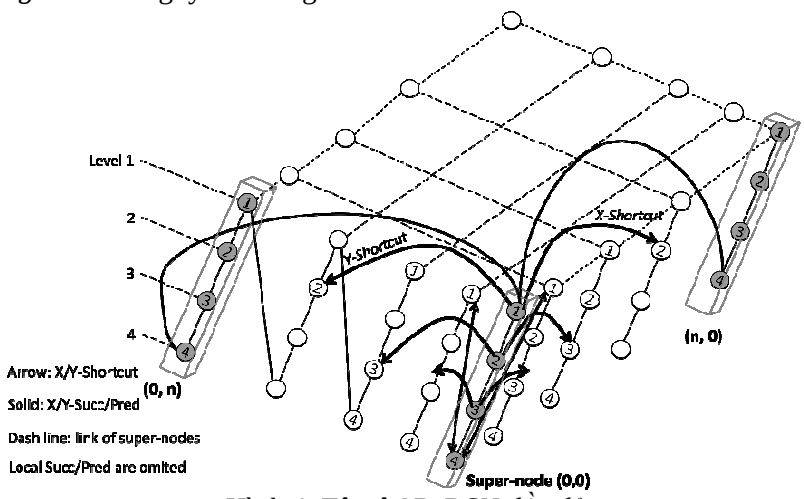
Loại liên kết *inter-cabinet* thứ 2 chính là các *Shortcut* được thiết kế một cách phù hợp nhằm giảm thiểu đường đi giữa các cabinet. Mỗi cabinet/super-node có *2p=2logn Shortcut* được phân phối như sau (Hình 3-b):

- Mỗi node thông thường có cấp từ *1* tới *p–1* có 2 *Shortcut gồm:*
  - *Shortcut* thứ nhất từ node (*X*, Y, *i*) tới node ($X + n/2^i$, Y, $i + 1$): đi tới super-node gần nhất ở khoảng cách $n/2^i$ trên trục X theo chiều kim đồng hồ; node đích có cấp $i + 1$.
  - *Shortcut* thứ hai từ node (X, *Y, i*) tới node (X, $Y + n/2^i$, $i + 1$): tương tự như trường hợp trên cho trục Y.
- Node cuối cùng của mỗi super-node với cấp *p* sẽ không có Shortcut.

Tổng kết lại cấu trúc mạng của 3D-DSN bao gồm *N* nodes, $N = n^2 \times log\, n$ với *n* nguyên dương. Mỗi node được đánh số với ID là *(X, Y, i)* trong đó *X, Y* = 1 … *n* và *i* = 1 … *log n*. Ngoài ra *(X, Y)* còn là ID của super-node tương ứng với node đó và *i* là cấp của node bên trong super-node.

Tô-pô 3D-DSN có tất cả 8 loại kết nối:

- *Intra-cabinet*: kết nối nội bộ cabinet; mỗi node có 2 kết nối nội bộ cabinet, gọi là *pred/succ***,** lần lượt tới node liền trước và liền sau theo chiều kim đồng hồ trong cùng cabinet.
- *Inter-cabinet*: kết nối liên cabinet được chia làm 2 nhóm
  - *Shortcut:* các node có cấp từ 1 tới *p – 1* có 2 *Shortcut:*
    - *X-shortcut*: từ node (*X*, Y, *i*) tới node ($X + n/2^i$, Y, $i + 1$)



Hình 4. Tô-pô 3D-DSN đầy đủ

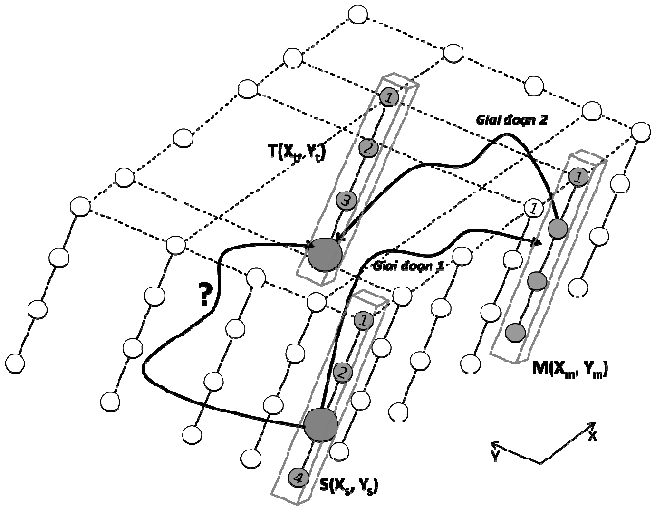
- *Y-shortcut*: từ node (X, *Y*, *i*) tới node (X, *Y* + $n/2^i$, *i* + *1*)

o Kết nối 2 cabinet liền kề:
- Node với cấp 1 có 2 kết nối *X-pred* và *Y-pred* lần lượt tới node có cấp cao nhất của cabinet liền trước theo chiều kim đồng hồ trên trục X, Y.
- Node với cấp p có 2 kết nối *X- succ* và *Y-succ* lần lượt tới node có cấp 1 của cabinet liền sau theo chiều kim đồng hồ trên trục X, Y.

Cấu trúc đầy đủ của 1 tô-pô 3D-DSN được thể hiện trên Hình 4.

IV.1.3. Giải thuật định tuyến

Giải thuật định tuyến của 3D-DSN được phát triển tự nhiên từ DSN trong đó điểm mấu chốt là cách sử dụng các *Shortcut* để *"cưa đôi"* đường đi từ một cabinet tới cabinet khác trên cùng trục X, Y. Đường đi từ node nguồn $s(X_s, Y_s, i_s)$ tới đích $t(X_t, Y_t, i_t)$ được chia làm 2 giai đoạn lớn là: giai đoạn một đi từ cabinet $S(X_s, Y_s)$ của node nguồn *s* tới cabinet trung gian $M(X_m, Y_m)$ sao cho *M* nằm trên cùng trục X so với cabinet nguồn *S*, nghĩa là: $Y_m = Y_s$, đồng thời *M* nằm trên cùng trục Y so với cabinet đích $T(X_t, Y_t)$, nghĩa là: $X_m = X_t$; giai đoạn hai đi từ cabinet trung gian *M* tới cabinet đích *T* (Hình 5). Mỗi giai đoạn này sẽ tương tự như một quá trình định tuyến giữa 2 node trên DSN.



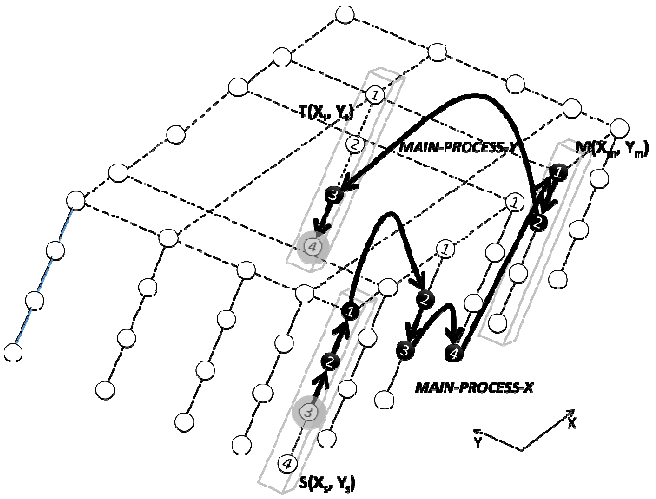
Hình 5. Đường đi cần tìm (dấu hỏi) được chia làm 2 giai đoạn bởi cabinet trung gian M

Ở giai đoạn thứ nhất, trước tiên giải thuật di chuyển nội bộ trong cabinet *S* để tìm node có cấp phù hợp sở hữu *X-Shortcut* có thể chia đôi đường đi từ cabinet *S* tới cabinet trung gian *M*, nghĩa là chiều dài của của *X-Shortcut* này nhỏ hơn khoảng cách (*S*, *M*) nhưng lớn hơn hoặc bằng 1/2 khoảng cách (*S*, *M*). Bằng cách liên tục sử dụng các *X-Shortcut* có đặc điểm này, sau mỗi lần giải thuật đảm bảo khoảng cách tới cabinet *M* được giảm đi một nửa (gọi là phương pháp chia đôi đường đi). Chú ý rằng các *Shortcut* trong giai đoạn này đều nằm trên cùng trục X nên quá trình này tương tự như phương pháp định tuyến trên ring của DSN. Hoàn toàn tương tự, giai đoạn thứ hai bắt đầu bằng việc tìm kiếm *Y-Shortcut* phù hợp trong nội bộ cabinet trung gian *M* nhằm cưa đôi đường đi tới cabinet đích *T*. Điểm khác biệt duy nhất là giai đoạn này hoàn toàn xảy ra trên trục Y với việc chọn lựa liên tiếp các *Y-Shortcut* để đi tới đích.

Chi tiết hơn, không mất tính tổng quá giả sử $X_t > X_s$ và $Y_t > Y_s$ và $d_X(S, T) = X_t - X_s$, $d_Y(S, T) = Y_t - Y_s$ là khoảng cách lần lượt trên trục X, Y giữa 2 cabinet *S*, *T*. Trước hết, trong giai đoạn 1 giải thuật sử dụng kết nối nội bộ (*succ* hoặc *pred*) để tìm kiếm node cấp *i* có *X-Shortcut* với độ dài $n/2^i$ phù hợp cho việc chia đôi đường đi trên trục X, nghĩa là: $\frac{1}{2}d_X(S,M) \le \frac{n}{2^i} \le d_X(S,M)$ hay $\lg n - \lg d_X(S,M) \le i \le \lg n - \lg d_X(S,M) + 1$, chú ý rằng nếu $\lg n - \lg d_X(S,M)$ là số nguyên ta chọn $i = \lg n - \lg d_X(S,M)$, ngược lại chọn $i = \lg n - \lg d_X(S,M) + 1$. Ta có thể hiểu rằng trong cabinet *S* đây là *X-Shortcut* dài nhất không vượt quá cabinet *M*. Sau khi chọn được *X-Shortcut* đầu tiên, ta có thể liên tục sử dụng các *X-Shortcut* ở chính node hiện tại (độ dài bằng 1/2 *Shortcut* vừa sử dụng), hoặc dùng các liên kết *succ* để tìm ra *X-Shortcut* ngắn phù hợp hơn ngay kế sau. Giai đoạn 2 khác biệt ở chỗ nó hoạt động trên trục Y, do đó cấp *i* phù hợp là: $i = \lg \mathrm{n} - \lg d_Y(S,M)$ hoặc $i = \lg n - \lg d_Y(S,M) + 1$. Giải thuật định tuyến giả ngôn ngữ chi tiết được trình bày trong phụ lục.

Giải thuật định tuyến được chia làm 5 pha nhỏ, trong đó 3 pha PRE-WORK, IMMEDIATE và

FINISH chỉ di chuyển nội bộ bên trong cabinet, 2 pha còn lại MAIN-PROCESS-X/Y là phần quan trọng nhất khi ta thực sự sử dụng các *Shortcut* để di chuyển giữa các cabinet. Chú ý rằng thủ tục nhỏ *ProperLevel(U, V, dimension)* được sử dụng để kiểm tra cấp phù hợp hiện tại trong cabinet *U* để sử dụng *Shortcut* đi tới cabinent *V* theo chiều *"dimension"*. Một đường đi điển hình được giải thuật tìm ra sẽ có dạng như trong Hình 6.



Hình 6. Đường đi điển hình trong 3D-DSN

IV.2 Đặc điểm nổi bật của tô-pô 3D-DSN

Trong phần này, chúng ta sẽ cùng điểm qua một số tính chất của tô-pô đề xuất 3D-DSN.

IV.2.1. Bậc hằng (Constant degree)

Tính chất 1: Tô-pô 3D-DSN có bậc (degree) hằng số mà cụ thể là 6, tương đương với 3D Torus.

Tính chất này giúp tiết kiệm các cổng mạng để kết nối được nhiều đơn vị tính toán hơn trên 1 switch, đồng thời cũng là tiết kiệm chi phí do chỉ phải mua các switch ít cổng hơn nhiều. Thậm chí khi đặt vào tình huống muốn sử dụng các switch rất nhiều cổng (32, 64), ngoài lợi ích trên ta còn có thể tận dụng các cổng để tăng cường kết nối cục bộ giữa các switch trong cùng cabinet với chi phí không đáng kể. Điều này là bởi vì để giảm được 1 liên kết trên đường đi giữa 2 node, ta có thể kết nối các swith trong cùng cabinet với độ dài dây ngắn sử dụng các cáp đồng giá rẻ thay vì sử dụng cáp quang đắt đỏ để kết nối 2 cabinet ở xa (vẫn là giảm 1 liên kết).

IV.2.2. Đường kính mạng

Tính chất 2: 3D-DSN có đường kính mạng logarithmic, nghĩa là *logN* trong đó *N* là kích thước mạng.

Chứng minh: Dựa vào giải thuật định tuyến trên, chúng ta sẽ tính toán để chỉ ra rằng đường đi tìm được giữa bất kỳ 2 node nào cũng là *logN*. Dễ dàng nhận thấy trong các pha PRE-WORK, IMMEDIATE và FINISH, giải thuật chỉ cần sử dụng các liên kết nội bộ cabinet *pred/succ* để tìm tới node mong muốn. Vì mỗi cabinet có $p = log\ n$ node thông thường, khoảng cách tối đa cần đi ở mỗi pha sẽ là $\frac{p}{2} = \frac{log\ n}{2}$, và tổng cộng 3 pha là $3/2\ logn$. 2 pha MAIN-PROCESS-X/Y hoàn toàn tương tự nhau ở cách sử dụng các *Shortcut* để chia đôi đường đi tới cabinet mong muốn. Ta có thể dễ dàng chứng minh rằng trong các pha này ([1]) ta luôn duy trì bất đẳng thức sau: $\frac{d_X(U,M)}{2} \leq \frac{n}{2^{l_u}} \leq d_X(U,M)$ đối với MAIN-PROCESS-X và $\frac{d_Y(U,T)}{2} \leq \frac{n}{2^{l_u}} \leq d_Y(U,T)$ đối với MAIN-PROCESS-Y. Mỗi pha kết thúc khi đã tới được cabinet mong muốn hoặc ở ngay liền trước (khi ở cấp cao nhất $l_u = p$, và bất đẳng thức trên đảm bảo ta đã tới ngay trước cabinet đích). Chú ý rằng trong 2 pha này, mỗi liên kết ta chọn dù là *X/Y-Shortcut* hay *X/Y-Succ* thì cấp đều tăng thêm 1 bậc. Vì có tất cả *p* cấp nên số liên kết tối đa cần cho mỗi pha này là $p = log\ n$. Tóm lại, tất cả 5 pha cần tối đa $\frac{3\ log\ n}{2} + 2\ log\ n = 3.5\ log\ n \approx 1.75\ log\ N$.

Thực tế cho thấy đây dĩ nhiên chưa phải là đường đi tối ưu. Tuy nhiên, nhờ vào giải thuật định tuyến này mà ta có một chặn trên cho đường kính mạng, là một thước đo đảm bảo rằng 3D-DSN có đường đi giữa các node nhỏ, do đó đảm bảo yêu cầu đỗ trễ mạng thấp. Ngoài ra kết quả lý thuyết này là tốt hơn so với kết quả tương tự trong [1], ở đó chặn trên được chỉ ra là 2.5*logN*, lớn hơn giá trị 1.75 *logN* ở đây, cũng gợi ý cho thấy đường kính của đồ thị 3D-DSN là nhỏ hơn so với DSN cơ bản.

IV.2.3. Chiều dài dây cáp, so sánh với 3D Torus

Tính chất 3: 3D-DSN và 3D Torus có cùng độ dài dây intra-cabinet, và lần lượt có độ dài dây inter-cabinet là $\frac{8}{3}n^3$, $2n^2 \log n$.

V. ĐÁNH GIÁ THỰC NGHIỆM

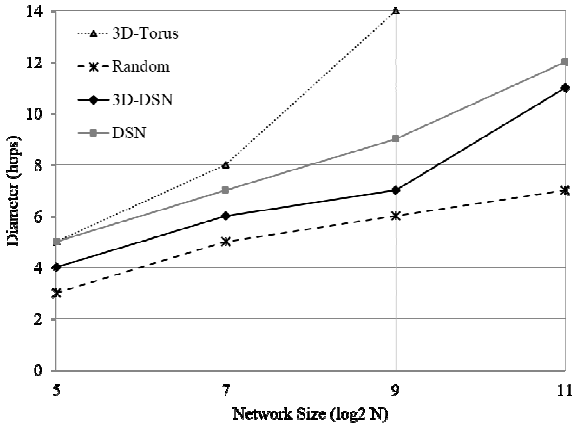
Trong phần này, chúng tôi trước hết so sánh đường kính và giá trị trung bình của đường đi ngắn nhất (average shortest path length -- ASPL) của tô-pô mạng liên kết 3D-DSN với các mạng liên kết điển hình. Sau đó, chúng tôi tính toán giá trị trung bình của độ dài cáp mạng khi thiết lập các mạng liên kết này vào trong phòng máy sử dụng cấu hình cài đặt của các hệ thống siêu máy tính hiện nay.

Mạng liên kết do chúng tôi đề xuất có bậc của đỉnh là hằng số và bằng 6. Theo đó, chúng tôi sẽ so sánh mạng liên kết này với các mạng liên kết khác có cùng số lượng nút mạng và cùng bậc. bao gồm tô pô truyền thống 3D-Torus, mạng liên kết ngẫu nhiên DLN-2-4 đề xuất trong [2] và mạng DSN cơ bản đề xuất trong [1]. Chúng tôi sẽ sử dụng các tên gọi ký hiệu sau lần lượt với các mạng liên kết đề cập trên: 3D-DSN, 3D-TORUS, RANDOM, và DSN.

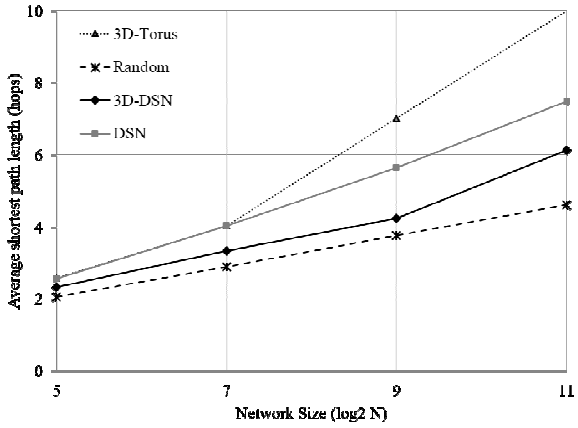
V.1. Đường kính và giá trị trung bình đường đi ngắn nhất

Hình 7 và Hình 8 thể hiện mối tương quan giữa đường kính mạng, độ dài đường đi ngắn nhất với kích thước mạng. Trong tất cả các kích thước mạng, mạng liên kết RANDOM có đường kính và độ dài đường đi ngắn nhất là nhỏ nhất. DSN và 3D-DSN lần lượt có kết quả tốt hơn Torus và tiếp cận dần tới RANDOM. Theo đó, mạng liên kết 3D-DSN được kì vọng có thể có đạt được hiệu năng (độ trễ truyền tin) gần giống với mạng liên kết RANDOM.

Mặt khác, quan sát Hình 7 và 8 cũng cho thấy khi kích thước mạng trở nên lớn hơn, cả RANDOM và 3D-DSN đều có đường kính và trung bình độ dài đường đi ngắn nhất là ngắn hơn so với DSN, và rất nhỏ so với Torus. Do đó chúng tôi nói rằng, hai mạng liên kết RANDOM và 3D-DSN này có khả năng mở rộng (scalability) tốt hơn Torus.



Hình 7. Diameter vs. Network Size



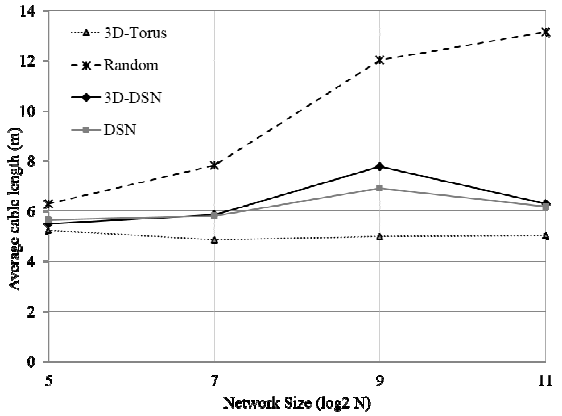
Hình 8. ASPL vs. Network size

Chúng tôi ước lượng độ dài cáp mạng cần thiết khi triển khai các mạng liên kết vào phòng máy vật lý của các tủ mạng (cabinet). Ở đó, diện tích của phòng máy được giả sử là không giới hạn nhằm phục vụ cho việc nghiên cứu các mạng liên kết có kích thước bất kì.

V.2. Trung bình độ dài cáp mạng

Các tủ mạng chứa cùng số lượng nút mạng và được đặt trong một lưới *AxB*. Trong đó $A = \lceil\sqrt{m}\rceil$ và $B = \lceil m/A \rceil$ với *m* là số lượng tủ mạng. Khi triển khai mạng liên kết 3D-DSN, mỗi tủ mạng trong phòng máy tương ứng chính xác với một siêu nút (super-node) trong mô hình thiết kế logic. Ví dụ, mạng 3D-DSN gồm *1024* nút mạng được cài đặt thành mạng lưới của *16x16* tủ mạng mà ở đó các tủ mạng cùng chứa 4 nút mạng. Cách sắp xếp bố trí tủ mạng này (layout) sau đó được sử dụng để triển khai các mạng liên kết còn lại trong quá trình đánh giá độ dài cáp mạng.

Với một cách bố trí phòng máy cụ thể, chúng tôi tính toán độ dài cáp mạng dựa trên trình bày trong [13]: độ dài dây cáp trung bình trong 1 tủ mạng là 2m, mỗi một dây cáp mạng được tính thêm một khoảng dôi ra (overhead) trung bình là 2m. Dây cáp nối các tủ mạng với nhau được tính dựa trên kích thước của tủ mạng. Chúng tôi cũng giả sử rằng, mỗi tủ mạng có kích thước 0.6m x 2.1m bao gồm cả không gian để cài đặt hệ thống làm mát (theo khuyến cáo trong [21]).



Hình 9. Average cable length vs. Network size

Hình 9 mô tả sự thay đổi của trung bình độ dài cáp mạng khi thay đổi kích thước của mạng theo từng mạng liên kết được chúng tôi đánh giá. Ở đó, các điểm có giá trị nhỏ hơn được đánh giá là tốt hơn trong đánh giá hiệu năng cho mạng liên kết. Nhận thấy rằng, 3D-DSN có kết quả tương tự như DSN trong hầu hết các trường hợp và nằm rất gần Torus (trường hợp tốt nhất). 3D-DSN cũng tiêu tốn độ dài cáp mạng ít hơn RANDOM và tăng dần từ 12.74% đến 42.03% theo sự tăng dần của kích thước mạng.

VI. KẾT LUẬN

Phát triển từ mạng DSN cơ bản [1], chúng tôi đã xây dựng tô-pô mạng liên kết 3 chiều 3D-DSN, tiếp tục phát huy điểm mạnh về tiết kiệm cáp mạng. Ý tưởng mở rộng của chúng tôi là lấy một 2D-Torus làm cơ sở thay vì 1D-Ring như trong [1]. Việc xây dựng các liên kết xa vì thế được tiến hành theo 2 phương, đồng thời thuật toán định tuyến được phát triển theo 5 pha thay vì 3 pha như trong [1]. Sở dĩ số pha tăng lên là do trong 3D-DSN, chúng tôi tách quá trình định tuyến theo siêu nút thành 2 giai đoạn cơ bản: đi theo chiều ngang rồi mới đi theo chiều dọc (tức là mỗi pha trong hai pha đầu của DSN – phục vụ quá trình định tuyến siêu nút – được tách thành 2 pha con trong 3D-DSN).

Chúng tôi tiến hành khảo sát các thuộc tính của tô-pô đề xuất trên cả 2 phương diện lý thuyết và thực nghiệm và đạt được các kết quả khả quan, cho thấy 3D-DSN là gần gũi với ứng dụng thực tế hơn so với DSN cơ bản. Về mặt lý thuyết (IV.2.2), thuật toán định tuyến của 3D-DSN có thể cho đường đi ngắn hơn so với DSN. Có thể nói 3D-DSN là một giải pháp trung gian hiệu quả giữa Torus và mạng ngẫu nhiên đề xuất bởi [2], cho phép cân bằng hiệu quả giữa đường kính đồ thị (và độ dài trung bình đường đi ngắn nhất) với chi phí cáp mạng.

LỜI CẢM ƠN

Chúng tôi xin cảm ơn các tác giả trong [17,19,20] đã đóng góp nhiều ý kiến gợi ý.

TÀI LIỆU THAM KHẢO

[1] VAN K. NGUYEN, N. T. X. LE, I. FUJIWARA, M. KOIBUCHI, "*Distributed Shortcut Networks: Layout-aware Low-degree Topologies Exploiting Small-world Effect*", In Proc. of The 42nd IEEE International Conference on Parallel Processing, Oct. Lyon, France (ICPP-2013)

[2] M. KOIBUCHI, H. MATSUTANI, H. AMANO, D. FRANK HSU, H. CASANOVA, "*A Case for Random Shortcut Topologies for HPC Interconnects*", In Proc. of the 39th International Symposium on Computer Architecture (ISCA'2012)

[3] O. LYSNE, L. PEDRO, M. KOIBUCHI, T. ROKICKI, & C. SANCHO, "*A Survey and Evaluation of Tô-pô Agnostic Deterministic Routing Algorithms*," IEEE Transactions on Parallel and Distributed Systems, 1-20. 2011

[4] J. KIM, W. J. DALLY, S. SCOTT, and D. ABTS, "Technology-Driven, Highly-Scalable Dragonfly Tô-pô," *in Proc. of the International Symposium on Computer Architecture (ISCA)*, 2008, pp. 77–88.

[5] P. COTEUS and et. al., "*Packaging the Blue Gene/L supercomputer*," IBM Journal of Research and Development, vol. 49, no. 2/3, pp. 213–248, Mar/May 2005.c

[6] Y. AJIMA, S. SUMIMOTO, and T. SHIMIZU, "*Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers,*" IEEE Computer, vol. 42, pp. 36–40, 2009.

[7] I. FUJIWARA, M. KOIBUCHI, and H. CASANOVA, "*Cabinet Layout Optimiza-tion of Supercomputer Topologies for Shor ter Cable Length,*" in Proc. of International Conference on Parallel and Distributed Com puting, Applications and Technologies, Dec 2012.

[8] Top 500 Supercomputer Sites, http://www.top500.org/.

[9] A. SINGLA, C.-Y. HONG, L. POPA, and P. B. GODFREY, "*Jellyfish: Net-working Data Centers Randomly,*" in Proc. of USENIX Symposium on Network Design and Implementation (NSDI), 2012.

[10] J. Y. SHIN, B. WONG, and E. G. SIRER, "*Small-World Data Centers,*" in Proc. of the Symposium on Cloud Computing, Oct. 2011.

[11] M. KOIBUCHI, I. FUJIWARA, H. MATSUTANI, and H. CASANOVA, "*Layout-conscious random topologies for hpc off-chip interconnects,*" in 19th International Conference on High-Performance Computer Architecture (HPCA), Feb. 2013, p. XX.

[12] JOURAKU, M. KOIBUCHI & H. AMANO, "*An effective design of deadlock-free routing algorithms based on 2d turn model for irregular networks,*" Parallel and Distributed Systems, IEEE Transactions on, 18(3), 320–333. IEEE. 2007

[13] J. KIM, W. J. DALLY and D. ABTS, "*Flattened Butterfly: A Cost-Efficient Topology for High-Radix Networks,*" Proceedings of the International Symposium on Computer Architecture (ISCA), pp. 126-137, 2 May 2007.

[14] D. WATTS and S. STROGATZ, "*Collective dynamics of small-world net-works,*" Nature, vol. 393, pp. 440–32, 1998.

[15] J. KLEINBERG, "*The small-world phenomenon: An algorithmic perspective,*" in STOC, 2000.

[16] C. MARTEL and V. NGUYEN, "*Analyzing Kleinberg's (and other) smallworld models,*" in PODC, 2004.

[17] M. R. SAMATHAM and D. K. PRADHAN, "*The De Bruijn Multiprocessor Network: A Versatile Parallel Processing and Sorting Network for VLSI,*" IEEE Trans. on Computers, vol. 38, no. 4, pp. 567–581, 1989.

[18] S. B. AKERS, B. KRISHNAMURTHY, and D. HAREL, "*The Star Graph: An Attractive Alternative to the n-Cube,*" in Proc. of the International Conference on Parallel Processing (ICPP), 1987, pp. 393–400.

[19] K. HWANG and J. GHOSH, "*Hypernet: A communication-efficient archi-tecture for constructing massively parallel computers,*" IEEE Trans. On Computers, vol. 36, no. 12, pp. 1450–1466, 1987.

[20] D. J. WATTS and S. H. STROGATZ, "*Collective dynamics of 'small-world'networks,*" Nature, vol. 393, no. 6684, pp. 440–442, 1998.

[21] HP, "*Optimizing facility operation in high density data center environments, technoloogy brief,*" [Online]. Available: http://h10032.www1.hp.com/ctg/Manual/c00064724.pdf

Nhận bài ngày: 19/4/2015

SƠ LƯỢC VỀ TÁC GIẢ

LÊ XUÂN THỐNG NHẤT

Sinh năm 1989.

Tốt nghiệp Đại học Bách Khoa Hà Nội năm 2012.

Hiện tại là nghiên cứu sinh tiến sỹ năm thứ nhất tại Database Lab, đại học California, Riverside, Hoa Kỳ.

Lĩnh vực nghiên cứu: Khai phá dữ liệu và mô hình mạng máy tính.

Email: nle020@ucr.edu

NGUYỄN KHANH VĂN

Sinh năm 1970 tại Hà nội.

Tốt nghiệp Trường ĐH Bách khoa Hà nội năm 1992. Nhận bằng Thạc sỹ năm 2000 tại ĐH Wollogong, Úc. Nhận bằng Tiến sỹ năm 2006 tại ĐH California-Davis, Hoa kỳ.

Hiện công tác tại Viện CNTT và Truyền thông, Trường ĐH Bách khoa Hà Nội.

Lĩnh vực nghiên cứu: Tính toán phân tán, Mô hình mạng mới và thuật toán, Mạng máy tính, An toàn thông tin.

Email: vannk@soict.hust.edu.vn