BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

***

ĐẶNG HÙNG VĨ

GIẢI PHÁP ĐIỀU KHIỂN CUNG CẤP TÀI NGUYÊN CHO HỆ PHÂN TÁN TRONG MÁY ẢO DỰA TRÊN KỸ THUẬT MÃ MẠNG

LUẬN ÁN TIẾN SĨ KỸ THUẬT

ĐÀ NẴNG, 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

***

ĐẶNG HÙNG VĨ

GIẢI PHÁP ĐIỀU KHIỂN CUNG CẤP TÀI NGUYÊN CHO HỆ PHÂN TÁN TRONG MÁY ẢO DỰA TRÊN KỸ THUẬT MÃ MẠNG

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ: 62.48.01.01

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Người hướng dẫn khoa học:
1. PGS.TS. Lê Văn Sơn
2. PGS.TSKH. Nguyễn Xuân Huy

ĐÀ NẴNG, 2020

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu do tôi thực hiện dưới sự hướng dẫn khoa học của PGS.TS. Lê Văn Sơn và PGS.TSKH. Nguyễn Xuân Huy. Các số liệu và kết quả trình bày trong luận án là trung thực, chưa được công bố bởi bất kỳ tác giả nào hay ở bất kỳ công trình nào khác.

Nghiên cứu sinh

Đặng Hùng Vĩ

MỤC LỤC

Lời cam đoan **i**
Mục lục **ii**
Danh mục các ký hiệu **iv**
Danh mục các từ viết tắt **vi**
Danh mục bảng, biểu **viii**
Danh mục hình vẽ **ix**
MỞ ĐẦU **1**
Chương 1. Tổng quan về cung cấp tài nguyên **7**
1.1 Điều khiển cung cấp tài nguyên 8
1.1.1 Các khái niệm và vấn đề cơ sở của tài nguyên 8
1.1.2 Nguyên lý và giải pháp điều khiển cung cấp tài nguyên 9
1.1.3 Bộ cung cấp tài nguyên 13
1.2 Các nghiên cứu liên quan 16
1.2.1 Các nghiên cứu liên quan đến điều khiển cung cấp tài nguyên trong hệ phân tán 16
1.2.2 Các nghiên cứu liên quan đến điều khiển cung cấp tài nguyên trong hệ thống ảo hóa 35
1.3 Mô hình và giải pháp điều khiển cung cấp tài nguyên trong hệ thống máy chủ ảo 56
1.3.1 Giới thiệu bài toán 57
1.3.2 Mô hình tổng quát 62
1.3.3 Giải pháp kỹ thuật 64
Tiểu kết Chương 1 70
Chương 2. Giải pháp điều khiển cung cấp tài nguyên truyền thông trong hệ phân tán **72**
2.1 Giải pháp song song hóa thuật toán Lamport trong loại trừ tương hỗ phân tán 72
2.1.1 Song song hóa trong thuật toán Lamport 72
2.1.2 Áp dụng song song hóa thuật toán Lamport để giải quyết loại trừ tương hỗ phân tán 77
2.1.3 Hiệu năng thực thi song song hóa thuật toán Lamport 78

2.2 Đề xuất thuật toán 4PCoDT điều khiển cung cấp tài nguyên trong hệ phân tán triển khai trong máy ảo 82
2.3 Triển khai giải pháp gắn bó trong hệ phân tán 86
2.3.1 Các hoạt động hệ phân tán 88
2.3.2 Triển khai thuật toán 4PCoDT trong hệ phân tán 91
2.3.3 Đánh giá và nhận xét các thuật toán trong hệ phân tán 93
Tiểu kết Chương 2 95
Chương 3. Kỹ thuật mã mạng tối ưu điều khiển cung cấp tài nguyên truyền thông trong hệ phân tán 96
3.1 Giải pháp điều khiển tỷ lệ nguồn với mã mạng 96
3.1.1 Các ràng buộc trong giải pháp cơ bản của kỹ thuật mã mạng . 96
3.1.2 Xác định các tỷ lệ và tối ưu hóa điều khiển tỷ lệ với đồ thị con . 98
3.2 Giải pháp tối ưu truyền thông multicast với mã mạng 107
3.2.1 Các yêu cầu về thông lượng và xây dựng tô pô mạng 107
3.2.2 Các kỹ thuật xử lý dòng thông tin 111
3.2.3 Xác định tỷ lệ lưu lượng trong cây multicast với mã mạng . . . 123
3.3 Đề xuất thuật toán xác định mã mạng tuyến tính 123
3.4 Đánh giá và nhận xét giải pháp kỹ thuật mã mạng cung cấp tài nguyên truyền thông cho hệ phân tán triển khai trong hệ thống ảo hóa 132
Tiểu kết Chương 3 133
KẾT LUẬN 135
DANH MỤC CÁC CÔNG TRÌNH CỦA TÁC GIẢ 138
Tài liệu tham khảo 139

DANH MỤC CÁC KÝ HIỆU

Ký hiệu	Diễn giải ý nghĩa
$suc[S_i]$	Máy chủ hàng xóm bên phải của S_i
$pred[S_i]$	Máy chủ hàng xóm bên trái của S_i
yc_i	Yêu cầu tài nguyên dùng chung thứ i
$\mathbb{YC}$	Tập yêu cầu tài nguyên dùng chung
H_S	Công tơ trạm S
sk_a	Sự kiện a
E	Dấu của sự kiện
S_i	Máy chủ ảo thứ i
S_{ij}	Liên kết giữa hai nút S_i và S_j liền kề trong hệ thống
$G = (\mathbb{U}, \mathbb{V})$	Mạng mô tả dưới dạng đồ thị G
$\mathbb{U}$	Tập các nút S_i
$\mathbb{V}$	Tập các cạnh S_{ij}
$\mathbb{T}$	Tập các nút trung gian
$ts_{S_{ij}}$	Trọng số nguyên dương với mỗi S_{ij}
p	Một phiên truyền thông điệp trong hệ thống
$\mathbb{P}$	Tập các phiên truyền thông điệp trong hệ thống
ll^p	Lưu lượng của một phiên p truyền trong mạng
S_0	Nút nguồn
$S_{\mathbb{U}}$	Nút đích
tl	Tỷ lệ lưu lượng truyền thông
$\mathbb{TL}$	Tập tỷ lệ lưu lượng truyền thông
DB_i	Cơ sở dữ liệu thứ i
F_i	Trường điều khiển thứ i của thông điệp trong hệ phân tán
$\oplus$	Phép toán XOR
F	Galois Field
x_{ij}	Xác suất liên kết i, j
M_i	Thông điệp thứ i
$dt_{k,l}^{pd}$	Dòng thông tin đối với đích d của phiên p
$dv_{k,l}^{p}$	Dòng vật lý của phiên p
$du(\cdot)$	Lưu lượng unicast

Ký hiệu	Diễn giải ý nghĩa
x	Liên kết của của cây multicast
$\mathbb{X}^p$	Tập chứa các liên kết của của cây multicast
C_x^p	Cây chứa tập các liên kết $V_x \in \mathbb{V}$
MT	Ma trận multicast
$T^p(\cdot)$	Hàm tiện ích phiên p
L	Phương pháp Lagrangian
$tn_{v,x}^p$	Giá trị tắc nghẽn tại liên kết v đối với cây multicast C_x^p
$H(\cdot)$	Hàm tiện ích tính lưu lượng và giá trị tắc nghẽn
y_t, bk	Bước kích thước vô hướng dương
$thl(\mathbb{U})$	Thông lượng cực đại của mạng
$g(\mathbb{U})$	Số lượng gói tin trên thời gian
$db(\mathbb{U})$	Loại kết nối vùng của mạng
$\mathbb{V}_{tp}$	Tập các liên kết liên thành phần
$NM\,\{S_0, S_j\}$	Cặp nút trong truyền thông nhóm
$kn(\mathbb{U})$	Các cạnh kết nối giữa NM
$Q(\cdot)$	Hàm tiện ích Q
$cp_{S_{i,j}}$	Chi phí cho mỗi đơn vị lưu lượng trên liên kết (i, j)
$kc_{S_{i,j}}$	Khoảng cách cho mỗi đơn vị lưu lượng trên liên kết (i, j)
$v_O(S_y)$	Biểu thị cung ra (id, S_y, S_z)
$v_I(S_y)$	Biểu thị cung vào (id, S_x, S_y)
id	Định danh của cung
h	Các cạnh song song $v_1, ..., v_h$
$\mathbb{C}_t$	Tập các cạnh đã xử lý gần nhất trong mỗi đường dẫn cạnh rời rạc h trong $\mathbb{P}^t$
B_t	Ma trận $[h \times h]$
b_i	Nguồn nguyên thủy thứ i
$start(v)$	Biểu thị cạnh v bắt đầu
$\overrightarrow{m_v}$	Véc tơ mã hóa cục bộ
$\overrightarrow{b\,(v)}$	Véc tơ mã toàn cục

DANH MỤC CÁC TỪ VIẾT TẮT

TT	Viết tắt	Dạng đầy đủ	Diễn giải ý nghĩa
1	2PC	Two-Phase Commit Protocol	Giao thức giao dịch 2 pha
2	3PC	Three-Phase Commit Protocol	Giao thức giao dịch 3 pha
3	4PCoDT	Four-phase Commit of Distributed Transaction	Giao dịch phân tán bốn pha
4	API	Application Programming Interface	Giao diện lập trình ứng dụng
5	ARQ	Automatic Repeat reQuest	Phương thức điều khiển lỗi truyền dữ liệu
6	BV		Người bảo vệ
7	CaaS	Communications as a Service	Truyền thông như là một dịch vụ
8	CNTT		Công nghệ thông tin
9	CPU	Central Processing Unit	Bộ xử lý trung tâm
10	DaaS	Data as a Service	Dữ liệu như là một dịch vụ
11	DB	Database	Cơ sở dữ liệu
12	DBaaS	Database as a Service	Cơ sở dữ liệu như là một dịch vụ
13	DS	Distributed System	Hệ phân tán
14	FIFO	First In First Out	Cơ chế vào trước ra trước
15	HDD	Hard Disk Drive	Ổ đĩa cứng
16	IaaS	Infrastructure as a Service	Hạ tầng như là một dịch vụ
17	IP	Internet Protocol	Giao thức Liên mạng
18	LAN	Local Area Network	Mạng cục bộ
19	ML	Maximum Likelihood	Tiêu chuẩn dự đoán Maximum Likelihood
20	NAT	Network Address Translation	Biên dịch địa chỉ mạng
21	NC	Network coding	Mã mạng
22	NFV	Network Functions Virtualization	Ảo hóa chức năng mạng
23	NIC	Network Interface Card	Card giao tiếp mạng
24	NV	Network Virtualization	Ảo hóa mạng

TT	Viết tắt	Dạng đầy đủ	Diễn giải ý nghĩa
25	ONF	Open Networking Foundation	Tổ chức Open Network Foundation
26	OSI	Open Systems Interconnection Reference Model	Mô hình tham chiếu OSI
27	P2P	peer-to-peer	Mạng ngang hàng
28	pps	packets per second	Số gói trong một giây
29	RADS	Resource Allocator in Distributed System	Bộ cung cấp tài nguyên phân tán
30	RAM	Random Access Memory	Bộ nhớ cho phép truy xuất đọc-ghi ngẫu nhiên
31	RAS	Resource Allocation System	Hệ thống cung cấp tài nguyên
32	SaaS	Software as a Service	Phần mềm như là một dịch vụ
33	SDI	Software Defined Infrastructure	Hạ tầng định nghĩa bằng phần mềm
34	SDN	Software Defined Networking	Mạng điều khiển bằng phần mềm
35	TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
36	TCP/IP	Transmission Control Protocol/ Internet Protocol	Giao thức TCP/IP
37	VLAN	Virtual Local Area Network	Mạng cục bộ ảo
38	VM	Virtual Machine	Máy ảo
39	VMS	Vitual Machine Systems	Hệ thống máy ảo
40	VNIC	Virtual Network Interface Card	Card giao diện mạng ảo
41	VPN	Virtual Private Network	Mạng riêng ảo
42	VT		Vị trí bãi đỗ xe

DANH MỤC BẢNG, BIỂU

1.1 So sánh giữa hệ tập trung và hệ phân tán 16
1.2 Ưu và nhược điểm của hệ phân tán 17
1.3 Các bước thực hiện thuật toán di chuyển tuần tự 21
1.4 Hoạt động diễn ra trên các máy chủ trong trật tự từng phần 29
1.5 Hiệu năng của thuật toán dựa trên quyền 30
1.6 Phân tích, so sánh hiệu năng của các thuật toán loại trừ tương hỗ . . . 30
1.7 So sánh 2PC và 3PC 34
1.8 Gói *network* trong trung tâm dữ liệu của Cloudsim 40
1.9 So sánh chi tiết Cloud Server và Máy chủ vật lý 53
1.10 Thông điệp phát đi trong hệ thống 61
1.11 Sự không gắn bó giữa bốn người bảo vệ 61
1.12 Mã thăm dò có thể quan sát được 69
1.13 Đánh giá tổng quát các nhóm giải pháp 71
2.1 Hoạt động diễn ra trên các máy chủ trong trật tự tổng quát chặt chẽ . 78
2.2 So sánh hiệu năng của thuật toán Lamport cải tiến trong loại trừ tương hỗ phân tán 82
2.3 Nội dung các trường điều khiển trong thông điệp 84
2.4 Các sự kiện đối với nút trong mô phỏng phân tán 90
2.5 So sánh giao thức 2 pha, 3 pha và 4 pha 94
3.1 Kết quả giá trị dựa trên song song hóa thuật toán Ford Fulkerson . . . 120
3.2 Thời gian thực thi của mã mạng và truyền multicast 130
3.3 Kết quả thực thi tô pô với 3 phương thức truyền 131

DANH MỤC HÌNH VẼ

1.1 Mô hình tổng quan kết nối trong hệ phân tán [65] 11
1.2 Kiến trúc truyền thống (a) và kiến trúc ảo hóa (b) [9] 15
1.3 Tô pô các loại thuật toán truyền multicast [62] 19
1.4 Mô hình tuần tự thông điệp [104] 20
1.5 Nhãn thời gian thông điệp không theo trật tự 26
1.6 Loại trừ tương hỗ nhờ dấu [65] 27
1.7 Sự kẹt chéo của các tiến trình gây bế tắc [115] 28
1.8 Hành động giao thức 2 pha 2PC [99] 32
1.9 Hành động giao thức 3 pha 3PC [99] 33
1.10 Các chiến lược cung cấp tài nguyên trong Điện toán đám mây [131] . . 36
1.11 Mô hình ảo hóa mạng [125] 39
1.12 Cấu trúc của SDN [49] 41
1.13 Cấu trúc của Floodlight 42
1.14 Mối quan hệ giữa Bộ điều khiển Floodlight 43
1.15 Hệ phân tán được tổ chức như phần mềm trung gian. Tầng phần mềm trung gian trải rộng trên nhiều máy chủ khác nhau và mỗi ứng dụng được cung cấp cùng giao diện tương tác [124] 46
1.16 Bốn thực thể của hệ tin học phân tán [65] 48
1.17 Hệ thống ảo hóa VMWare 52
1.18 Cấu hình của một máy ảo thực thi hệ phân tán 54
1.19 Hệ phân tán triển khai trên hệ thống máy ảo 55
1.20 Sơ đồ tổng quát về hệ thống giám sát phương tiện cơ giới đường bộ . . 57
1.21 Kết nối truyền thông trong nước của các nhà cung cấp dịch vụ 58
1.22 Mô phỏng bài toán bãi đỗ xe [65] 60
1.23 Mô hình tổng quát cung cấp tài nguyên trong hệ thống máy ảo 63
1.24 Mạng truyền thông phân tán ảo 65
1.25 Hệ thống mạng biểu diễn dưới dạng đồ thị 65
1.26 Cách thức truyền unicast (a) và multicast (b) 68
1.27 Cách thức truyền multicast kết hợp với mã mạng 69
2.1 Trật tự tổng quát các thông điệp theo thuật toán Lamport sau khi cải tiến so với Hình 1.6 76
2.2 Cung cấp tài nguyên phân tán cho cặp yêu cầu/đáp ứng 77
2.3 Mô tả các tiến trình hoạt động trong miền găng 80

2.4 Thông điệp di chuyển theo vòng tròn ảo 83
2.5 Cấu trúc thông điệp di chuyển theo vòng tròn ảo của hệ phân tán . . . 84
2.6 Thuật toán 4PCoDT đảm bảo gắn bó trong hệ phân tán 85
2.7 DSSim đọc thông số tô pô 87
2.8 Các sự kiện trong chương trình mô phỏng DSSim 88
2.9 Giao diện thể hiện kết quả thực thi tô pô trên công cụ mô phỏng DSSim 89
2.10 Cấu trúc thông điệp trong hệ thống mô phỏng DSSim 91
2.11 Trạng thái các bảng dữ liệu trên các máy chủ 92
2.12 Chương trình thể hiện thuật toán 4PCoDT 93
3.1 Cây multicast 99
3.2 Mô tả gói tin được chia thành nhiều khối và mã hóa 105
3.3 Chia tỷ lệ và truyền gói tin đã mã từ Server đến Client 106
3.4 Tô pô các nút mạng ban đầu 112
3.5 Pha đầu trong thuật toán xóa liên kết 113
3.6 Thuật toán tối ưu cục bộ 116
3.7 Kết quả thuật toán thêm liên kết (a) và xóa liên kết (b) 117
3.8 Sơ đồ song song hóa thuật toán Ford Fulkerson 121
3.9 Tính toán luồng cực đại từ nguồn đến đích 122
3.10 Ưu điểm của mã mạng trong cải tiến thông lượng multicast từ nguồn đến tập đích 124
3.11 Mô hình mã mạng 124
3.12 Biểu thị cung vào/ra của nút S_y 125
3.13 Tỷ lệ truyền thông với 3 phương thức truyền 131

MỞ ĐẦU

1. Lý do chọn đề tài

Theo các tác giả Silberschatz, Galvin và Gagne [114] trình bày các khái niệm hệ điều hành, trong đó nêu khái niệm cung cấp và chia sẻ tài nguyên (phần cứng, phần mềm, dữ liệu, truyền thông) nhằm đảm bảo hoạt động thông suốt cho các tiến trình diễn ra bên trong máy tính. Tài nguyên hữu hạn nhưng nhu cầu sử dụng tài nguyên thì vô hạn. Nếu các tài nguyên không được giám sát, quản lý một cách hợp lý và có sự phân chia rõ ràng thì sẽ dẫn đến tương tranh, bế tắc và thiếu tài nguyên. Do đó, các chiến lược cung cấp tài nguyên được nghiên cứu và hình thành dựa trên các yếu tố này. Các hệ thống cung cấp tài nguyên, chương trình quản lý và giám sát được xem như là một phần của hệ điều hành máy tính.

Tác giả Peterson và Davie [102] trình bày một mạng máy tính bao gồm các máy tính, máy chủ được kết nối với nhau nhằm chia sẻ tài nguyên thông qua các thiết bị truyền thông. Trong một mạng máy tính, số lượng kênh truyền, băng thông và lưu lượng trong truyền thông được xét là một trong số các loại tài nguyên và được quan tâm nghiên cứu như các loại tài nguyên khác.

Dối với cung cấp tài nguyên truyền thông trong hệ tập trung, truyền thông điểm đến điểm (point-to-point) được sử dụng cho kết nối giữa các máy chủ và unicast là đặc trưng cho loại kết nối này. Theo các tác giả Kwok, Chan, Cheung [64], Serge và cộng sự [41], các máy chủ sử dụng giao thức kết nối P2P (Peer-to-Peer) với nhau và trao đổi thông điệp theo từng cặp. Do đó, hạn chế là khả năng mở rộng khi hệ thống được triển khai ở quy mô lớn.

Tác giả Peterson và Davie [102] nêu ra nhược điểm của hệ tập trung là xảy ra tắc nghẽn truyền thông tại máy chủ khi có quá nhiều thiết bị đầu cuối yêu cầu chia sẻ tài nguyên truyền thông dẫn đến hiệu năng cung cấp tài nguyên của hệ thống giảm. Ngoài ra, những nhược điểm khác của hệ tập trung như: sự cố hệ điều hành, phần mềm, phần cứng, đường truyền,... cũng ảnh hưởng đến hoạt động cung cấp tài nguyên. Bên cạnh đó, việc chia sẻ tài nguyên dùng chung xảy ra bế tắc trong quá trình cung cấp khi số lượng yêu cầu tài nguyên lớn hơn so với tài nguyên hiện có. Các nhược điểm của hệ tập trung đã được các nhà khoa học, nhà triển khai hệ thống quan tâm nghiên cứu, phát triển và xây dựng hệ phân tán (Distributed System).

Hệ phân tán áp dụng các giải pháp và thuật toán khắc phục các nhược điểm của hệ tập trung được trình bày bởi tác giả Lê [65], Tanenbaum [124], Haddad và cộng sự [41]. Để giải quyết nhược điểm trong truyền thông, hệ phân tán thay thế phương thức truyền unicast bằng phương thức truyền multicast. Chính vì sử dụng phương thức truyền multicast nên việc triển khai cung cấp tài nguyên dùng chung trong hệ phân tán phức tạp hơn so với triển khai trong hệ tập trung. Các công bố của tác giả Lê và cộng sự trình bày các giải pháp như: đảm bảo tính gắn bó trong hệ phân tán trong nghiên cứu [66, 68, 70] và xử lý song song tiến trình trong môi trường phân tán trong nghiên cứu [67] nhằm phòng, tránh bế tắc trong cung cấp tài nguyên dùng chung.

Theo các tác giả Cao và Singhal trong [10], Ensink và Adve trong [30] trình bày cung cấp tài nguyên dùng chung trong hệ phân tán, các thông điệp liên tục truyền giữa các máy chủ với nhau để thực hiện nhiệm vụ chung được gọi là *hợp lực* (coordinator). Bên cạnh đó, tác giả Singhal [115] trình bày trong quá trình *hợp lực*, nhiều tiến trình cùng tương tranh tài nguyên dùng chung dễ dẫn đến bế tắc trong cung cấp tài nguyên. Nếu không giải quyết tốt vấn đề cung cấp tài nguyên dùng chung, một tài nguyên có thể được cấp cho nhiều tiến trình khác nhau hoặc không cung cấp cho tiến trình nào; điều này dẫn đến thiếu gắn bó và phá vỡ hoạt động của hệ phân tán.

Các tác giả Carvalho và Roucairol [11], Lê [65], Nohamed và Michel [91], Suzuki và Kasami [120] trình bày *loại trừ lẫn nhau phân tán hay còn gọi là loại trừ tương hỗ phân tán* (distributed mutual exclusion) nhằm cho phép chỉ có một tiến trình duy nhất được thực thi trong *miền găng* (critical section) tại một thời điểm đối với một tài nguyên để giải quyết bế tắc. Các tác giả Lê [65], Tanenbum và Steen [124] trình bày *loại trừ tương hỗ phân tán* sử dụng cơ chế *truyền thông điệp* (message passing) để xử lý các tiến trình yêu cầu tài nguyên dùng chung.

Trong cơ chế truyền thông điệp, các tác giả Chiu và Hsiao [14], Défago và cộng sự [25], Kshemkalyani và Singhal [60, 61], Lamport [71], Raynal và cộng sự [105] đề cập đến việc xác định trật tự các tiến trình yêu cầu tài nguyên dùng chung trên các máy chủ được chú trọng bởi truyền thông có độ trễ dẫn đến *trật tự các tiến trình trên các máy chủ khác nhau*. Khi các tiến trình không nằm trên một trật tự thống nhất dẫn đến các xử lý trên dữ liệu sẽ khác nhau và không nhất quán dữ liệu giữa các máy chủ. Để đảm bảo trật tự trên các máy chủ, các thuật toán Lamport [71], Ricart-Agrawala [107], Raymond [103], Maekawa [83], Suzuki-Kasami [120], Naimi-Trehel [92] được áp dụng để loại trừ tương hỗ trong chương trình điều khiển phân tán trên các máy chủ. Ngoài ra, các tác giả Ozsu và Valduriez [99] trình bày chương trình phân tán

phối hợp với hệ quản trị cơ sở dữ liệu phân tán thông qua giao dịch phân tán ba pha (3PC) nhằm đảm bảo tính nhất quán dữ liệu.

Cơ chế truyền thông điệp nhân bản các thông điệp và truyền song song trên các kênh trên máy chủ. Các tác giả Ho và cộng sự [47], Li và cộng sự [78], Li [80] trình bày về vấn đề thông tin trong truyền thông nhiều máy chủ có nhược điểm là dư thừa khi truyền multicast cùng một thông điệp đến tập đích. Bên cạnh đó, các tác giả Kshemkalyani và Singhal [62] trình bày trong quá trình hợp lực để xác định giá trị đồng hồ lô gic để đảm bảo trật tự các tiến trình trên các máy chủ yêu cầu phải tiếp nhận đầy đủ thông điệp phản hồi mới được phép vào miền găng. Các tác giả Kshemkalyani và Singhal [62] nêu ra nhược điểm nếu một trong số máy chủ bị sự cố hoặc thất lạc gói tin truyền, như vậy các tiến trình phải chờ đợi dẫn đến giảm hiệu năng của hệ phân tán. Ngoài ra, tác giả Manoj [63] trình bày trong giao dịch 3PC có nhược điểm là chi phí khôi phục lớn nếu một trong các giao dịch bị thất bại, vấn đề này làm tăng độ phức tạp trong hệ phân tán.

Vì vậy, Luận án tập trung *nghiên cứu, xây dựng bộ cung cấp tài nguyên truyền thông nhằm tối ưu truyền thông điệp giữa các máy chủ và đảm bảo tính gắn bó trong hệ phân tán*. Theo các tác giả Chou và cộng sự [17], Koetter và Medard [57] trình bày ưu điểm của kỹ thuật *mạng mã hóa hay mã mạng* (Network Coding) để giải quyết vấn đề trùng lặp gói tin tại tập đích. Các tác giả Ho và cộng sự [47], Li và cộng sự [78], Xi và Yeh [137] trình bày ưu điểm triển khai thuật toán phân tán dựa trên truyền multicast với mã mạng. Do đó, giải pháp nghiên cứu của Luận án về điều khiển lưu lượng thông tin và định tuyến gói tin trong mạng dựa trên kỹ thuật mã mạng nhằm tối ưu trong quá trình truyền multicast.

Các tác giả Langberg và cộng sự [73], Medard và Sprintson [85] trình bày khái niệm về mã mạng là kỹ thuật tối ưu luồng dữ liệu số (digital data) trên mạng. Mã mạng không chuyển *các gói* (packet) của thông điệp qua mạng mà thay vào đó là chuyển các *dấu hiệu số* (digital evidence) của một hoặc nhiều thông điệp qua các đường truyền trên mạng. Tại nút đích, thông điệp cần nhận được suy ra từ *dấu hiệu số*, kể cả trường hợp một vài gói của thông điệp bị thất lạc trên đường truyền. Khi các bít của *dấu hiệu số* đến nút đích, thông điệp cần truyền sẽ được giải mã từ chính nó chứ không phải là quá trình hợp gói như theo cách thông thường, chính vì vậy tốc độ truyền và nhận thông điệp sẽ được cải thiện một cách đáng kể.

Mục đích của Luận án là *xây dựng bộ cung cấp tài nguyên* nhằm tránh dư thừa thông tin, điều khiển tỷ lệ nguồn và tối ưu truyền thông để đạt được tỷ lệ nhận cao tại tập đích và phòng tránh tắc nghẽn trong truyền thông. Các tác giả Chou và cộng

sự [17], Szabó và cộng sự [122, 121] trình bày bộ cung cấp tài nguyên được xây dựng và triển khai trong hệ thống ảo hóa nhằm tăng tính linh hoạt và mềm dẻo trong các hệ thống Điện toán đám mây.

Theo Velte và cộng sự [130] trình bày hệ thống Điện toán đám mây sử dụng các tài nguyên tính toán động, do đó chiến lược cung cấp tài nguyên mang tính chất phức tạp hơn so với cung cấp tài nguyên trong hệ tập trung và phân tán. Vì vậy, các nghiên cứu về cung cấp, tối ưu tài nguyên truyền thông đang là một trong những thách thức đối với các nhà nghiên cứu. Trong cung cấp tài nguyên truyền thông, các tác giả Chowdhury và Boutaba [18], Haider và cộng sự [42] trình bày ảo hóa mạng đang là một trong những vấn đề quan tâm và thách thức trong nghiên cứu và triển khai.

Các tác giả Feamster và cộng sự [34], Jain và Paul [51], Krigslund và cộng sự [59], Metzler và cộng sự [87] đã trình bày khái niệm *mạng được điều khiển bằng phần mềm* (SDN - Software defined networking) là phương pháp tiếp cận mới trong ảo hóa mạng. SDN là kiến trúc mạng trong Điện toán đám mây được thiết kế cho phép hệ thống mạng trở nên linh hoạt. SDN là kiến trúc cho phép tách riêng kiểm soát luồng mạng (control plane) với luồng dữ liệu (data plane) dựa trên giao thức luồng mở (Open Flow). SDN tách định tuyến và chuyển các luồng dữ liệu riêng rẽ và chuyển kiểm soát luồng sang thành phần mạng riêng có tên gọi là thiết bị kiểm soát luồng (Flow Controller). Điều này cho phép luồng các gói dữ liệu đi qua mạng được kiểm soát theo lập trình. Do đó, sự kết hợp mã mạng trong SDN khi cung cấp tài nguyên truyền thông cho hệ thống máy ảo có thể tối ưu trong quá trình truyền và giảm tắc nghẽn. Hướng nghiên cứu của Luận án nhằm mục đích *tối ưu truyền thông trong máy ảo dựa trên kỹ thuật mã mạng đảm bảo cung cấp tài nguyên truyền thông và tránh tắc nghẽn cho các ứng dụng hệ phân tán.*

Trên cơ sở các nghiên cứu và triển khai về cung cấp tài nguyên cho hệ phân tán trong máy ảo vẫn còn nhiều yếu tố kỹ thuật cần xây dựng và phát triển; *giải pháp điều khiển cung cấp tài nguyên cho hệ phân tán trong máy ảo dựa trên kỹ thuật mã mạng* là vấn đề cần nghiên cứu trong giai đoạn hiện nay.

2. Mục tiêu, đối tượng và phạm vi nghiên cứu

2.1. Mục tiêu nghiên cứu

Mục tiêu chính của Luận án nhằm giải quyết bài toán *điều khiển cung cấp tài nguyên cho hệ phân tán triển khai trong hệ thống ảo hóa dựa trên kỹ thuật mã mạng (network coding technique)*, mục tiêu cụ thể:

- Nghiên cứu, đề xuất giải pháp, thuật toán điều khiển cung cấp tài nguyên đảm

bảo tính gắn bó cho hệ phân tán thông qua cơ chế truyền thông điệp.

- Nghiên cứu, đề xuất giải pháp, thuật toán tối ưu và tránh bế tắc trong cung cấp tài nguyên truyền thông cho hệ phân tán triển khai trong máy ảo dựa trên kỹ thuật mã mạng.

2.2. Đối tượng và phạm vi nghiên cứu

Điện toán đám mây là môi trường hoạt động cung cấp dịch vụ bao gồm các thành phần: người sử dụng và các thiết bị đầu cuối, môi trường truyền thông, bộ điều khiển cung cấp tài nguyên, dịch vụ đám mây, hạ tầng vật lý. Mỗi thành phần có chức năng, vai trò khác nhau nhưng liên kết tạo thành một thực thể thống nhất không thể tách rời. Luận án tập trung nghiên cứu, xây dựng giải pháp đảm bảo cung cấp tài nguyên dùng chung cho người sử dụng trong hệ phân tán và tối ưu cung cấp tài nguyên truyền thông cho hệ phân tán trong hệ thống ảo hóa dựa trên kỹ thuật mã mạng.

3. Phương pháp nghiên cứu

Luận án được triển khai với các phương pháp nghiên cứu như sau:

- Phương pháp lý thuyết: tìm kiếm, thu thập tài liệu, kết quả nghiên cứu các công trình đã được công bố, các tạp chí, hội nghị, hội thảo trong và ngoài nước để đưa ra mô hình bài toán tổng quát và hướng giải quyết bài toán cho thành phần trong nghiên cứu.

- Phương pháp thực nghiệm khoa học: sử dụng các công cụ mô phỏng để thực nghiệm các giải pháp, thuật toán. Dựa trên các kết quả mô phỏng phân tích, đánh giá giải pháp, thuật toán đề xuất.

4. Ý nghĩa khoa học và thực tiễn

Những đóng góp chính của Luận án về mặt khoa học và thực tiễn như sau:

- Đề xuất song song hóa thuật toán Lamport và thuật toán 4PCoDT để cải tiến quá trình trao đổi thông điệp, đảm bảo tính gắn bó dựa trên giao dịch 4 pha trong hệ thống phân tán.

- Đề xuất hai thuật toán thêm liên kết và xóa liên kết nhằm rút gọn cây multicast. Cây multicast rút gọn kết hợp song song hóa thuật toán Ford-Fukerson tìm lưu lượng cực đại trong truyền multicast. Giải pháp truyền thông multicast kết hợp với mã mạng triển khai trong hệ thống máy ảo nhằm phòng tránh tắc nghẽn, tối ưu cung cấp tài nguyên truyền thông cho hệ phân tán.

5. Cấu trúc Luận án

Cấu trúc Luận án nhằm nêu rõ các phân tích, đánh giá các nghiên cứu đã công bố và giải quyết các vấn đề còn tồn tại, hạn chế dựa trên đề xuất các giải thuật, thuật toán. Ngoài phần mở đầu, kết luận và hướng phát triển, bố cục của Luận án được chia thành ba chương:

- Chương 1 trình bày tổng quan về cung cấp tài nguyên, các nguyên lý và điều khiển trong cung cấp tài nguyên. Bên cạnh đó, Chương 1 trình bày các nghiên cứu liên quan đến việc điều khiển cung cấp tài nguyên để từ đó rút ra mô hình, bài toán tổng quát cho Luận án về điều khiển cung cấp tài nguyên cho hệ phân tán triển khai trong máy ảo dựa trên kỹ thuật mã mạng.
- Chương 2 trình bày nghiên cứu giải pháp cung cấp tài nguyên trong hệ phân tán. Chương 2 đề cập song song hóa thuật toán Lamport nhằm đánh dấu và thiết lập trật tự tổng quát các tiến trình trong loại trừ tương hỗ. Áp dụng song song hóa thuật toán Lamport giải quyết tiến trình duy nhất đi vào miền găng trong loại trừ tương hỗ phân tán. Kết quả giải pháp cải tiến trong Chương 2 nhằm đảm bảo tính gắn bó trong hệ phân tán thông qua thuật toán 4PCoDT.
- Chương 3 trình bày giải pháp tối ưu cung cấp tài nguyên truyền thông trong hệ phân tán dựa trên kỹ thuật mã mạng. Giải pháp này triển khai trong hệ thống ảo hóa nhằm các mục đích điều khiển tỷ lệ nguồn với đồ thị con được mã hóa. Xây dựng giải pháp điều khiển tỷ lệ nguồn được phát triển trên thuật toán điều khiển thích nghi tỷ lệ tương ứng. Hai thuật toán tạo tô pô là thuật toán thêm liên kết và thuật toán xóa liên kết trong việc xây dựng tô pô truyền multicast kết hợp với mã mạng. Song song hóa thuật toán Ford-Fulkerson nhằm tìm luồng cực đại trong truyền multicast. Thuật toán xác định mã mạng đảm bảo tập đích nhận thông tin không bị trùng lặp và đạt được thông lượng cực đại.

Chương 1.

TỔNG QUAN VỀ CUNG CẤP TÀI NGUYÊN

Tài nguyên là một trong những thành phần được cung cấp cho các tiến trình trong các hệ thống tính toán. Nhu cầu sử dụng tài nguyên ngày càng tăng dẫn đến các thiết bị CNTT (máy tính, máy chủ, thiết bị truyền thông,...) phát triển ngày càng đa dạng và phức tạp. Sự phát triển của tài nguyên kéo theo việc cải tiến các chiến lược nhằm tối ưu và phòng, tránh bế tắc trong cung cấp tài nguyên.

Hệ phân tán được quan tâm nghiên cứu và triển khai để giải quyết các hạn chế của hệ tập trung. Hệ phân tán hoạt động dựa vào cơ chế *truyền thông điệp* để *hợp lực* giữa các máy chủ với nhau nhằm cung cấp tài nguyên dùng chung. Điều này cho thấy, môi trường truyền thông đóng vai trò quan trọng trong hệ phân tán. Mặc dù, khắc phục được các hạn chế của hệ tập trung nhưng hệ phân tán gặp phải nhược điểm là *chi phí cho tài nguyên vật lý lớn* và *giải pháp điều khiển cung cấp tài nguyên phức tạp.*

Cuộc cách mạng về cung cấp tài nguyên tạo nên một sự bức phá đó là sự hình thành lĩnh vực *Điện toán đám mây.* Điện toán đám mây tạo ra một sự thay đổi cơ bản trong kiến trúc máy tính, phát triển phần mềm, các công cụ và trong cách lưu trữ, sử dụng, phân tán tài nguyên. Tính linh hoạt của Điện toán đám mây thể hiện rõ thông qua bộ cung cấp tài nguyên theo yêu cầu. Bộ cung cấp tài nguyên tạo điều kiện thuận lợi cho việc sử dụng các tài nguyên sẵn có của hệ thống, các tài nguyên được sử dụng như một máy tính gộp ảo. Cấu hình hợp nhất của Điện toán đám mây cung cấp một môi trường ở đó các ứng dụng thực hiện một cách độc lập mà không quan tâm đến bất kỳ cấu hình cụ thể nào.

Việc cung cấp tài nguyên là vấn đề đang được quan tâm nghiên cứu và triển khai trên các hệ thống nói chung, hệ thống Điện toán đám mây nói riêng. Vì vậy, mô tả tài nguyên trong pha hình thành là một trong những việc cần thiết để xây dựng các chiến lược điều khiển cung cấp tài nguyên trong pha vận hành. Để đạt được mục đích tối ưu trong cung cấp tài nguyên, tài nguyên cần phải được điều khiển dựa vào các nguyên lý và giải pháp cho từng loại cụ thể.

1.1. Điều khiển cung cấp tài nguyên

1.1.1. Các khái niệm và vấn đề cơ sở của tài nguyên

Theo Mục 1.1, Phần 1 của tác giả Abraham trong [114] trình bày khái niệm: *Tài nguyên máy tính* là sự tập hợp các nguồn có sẵn như phần cứng, phần mềm, tài liệu, thiết bị, các dịch vụ hỗ trợ và người sử dụng. Theo nghiên cứu các tác giả Frank, Kahn và Kleinrock trong [36] trình bày khái niệm: *Chia sẻ tài nguyên* hay còn gọi là *Tài nguyên truyền thông* tham chiếu đến dạng dữ liệu, thông tin và các thiết bị phần cứng dễ dàng truy cập bởi máy tính từ xa thông qua mạng cục bộ. Theo Tanenbaum và Steen [124], *tài nguyên dùng chung* đóng vai trò quan trọng trong cung cấp ứng dụng người sử dụng, trong đó hợp lực giữa các máy chủ là cần thiết để đảm bảo tính gắn bó. Theo Mell và Grance trong [86] đưa ra định nghĩa: Điện toán đám mây là mô hình cho phép truy cập phổ biến, dễ dàng và theo yêu cầu đến một nhóm *tài nguyên Điện toán đám mây* như mạng, máy chủ, lưu trữ, ứng dụng, dịch vụ,... đã được cấu hình và được quản lý hiệu quả trong cung cấp, giải phóng tài nguyên một cách nhanh chóng. Các giải pháp Điện toán đám mây cung cấp tài nguyên cho người sử dụng và doanh nghiệp có khả năng khác nhau để lưu trữ và xử lý dữ liệu của họ trong các trung tâm dữ liệu [39, 128, 131, 112, 33, 84, 100, 117, 139]. Giải pháp Điện toán đám mây dựa vào tài nguyên dùng chung để đạt được sự gắn bó với quy mô lớn, tương tự như một tiện ích qua mạng.

Vì vậy, khái niệm tổng quát về tài nguyên được Luận án khái quát hóa như sau: *Tài nguyên là các nguồn sẵn có (phần cứng, phần mềm, dữ liệu, băng thông, thiết bị vào/ra, dịch vụ,...) cung cấp cho máy tính hoạt động và cung cấp hoạt động ứng dụng cho người sử dụng.* Khái niệm về cung cấp tài nguyên của Luận án được phát biểu như sau: *Cung cấp tài nguyên đóng vai trò điều phối, điều khiển tài nguyên sẵn có nhằm tối ưu cho nhiều mục đích khác nhau thông qua các giải thuật, thuật toán và chiến lược cung cấp.*

Trong việc cung cấp tài nguyên, cần thiết phải mô tả tài nguyên để từ đó có các chiến lược để đạt được mục đích trong cung cấp tài nguyên. Các vấn đề cơ sở và phương pháp tiếp cận để cung cấp tài nguyên đó là: cung cấp thủ công, sử dụng các thuật toán để cung cấp tự động và kết hợp cả hai. Trong quá trình cung cấp, hiệu năng của hệ thống cung cấp là tài nguyên phải được *xử lý tối ưu* và *tránh tương tranh.*

Theo Haddad, Kordon và Petrucci [41], *tối ưu tài nguyên* dựa trên các giải thuật, thuật toán nhằm cho phép tài nguyên được cung cấp một cách hiệu quả nhất để đảm bảo hiệu năng của hệ thống đạt được cao nhất có thể. Mục đích của tối ưu cung cấp

tài nguyên trong hệ tập trung dựa vào ba yếu tố: thực thi tiến trình nhanh nhất có thể, chi phí xử lý thấp nhất và không vượt quá nguồn tài nguyên sẵn có

Trong cung cấp tài nguyên, một trong những vấn đề quan tâm của các nhà nghiên cứu và triển khai ứng dụng đó là *tương tranh (concurrency) tài nguyên*. Bộ cung cấp tài nguyên không giải quyết được tương tranh triệt để dẫn đến *bế tắc (deadlock)* và *thiếu tài nguyên*. Theo các tác giả Tanenbaum, Steen [124], Haddad, Kordon, Petrucci [41] và Kshemkalyani, Singhal [60], *tương tranh tài nguyên* là trạng thái số lượng lớn tiến trình xuất phát từ các ứng dụng độc lập muốn truy cập vào các tài nguyên với số lượng có hạn hay truy cập vào thông tin dùng chung cùng một lúc. Bên cạnh đó, các tác giả cũng nêu vấn đề *bế tắc* hay còn gọi là *khóa tương hỗ (mutual)* là sự kẹt chéo lẫn nhau của các tiến trình. *Bế tắc* diễn ra khi hai tiến trình đang sử dụng hai tài nguyên lại phát yêu cầu về nhu cầu sử dụng tài nguyên mà tiến trình kia còn đang sử dụng. *Thiếu tài nguyên* là sự chờ đợi quá lâu của một tiến trình mà yêu cầu của nó trễ đến mức không thể xác định được.

Hai vấn đề *tối ưu tài nguyên*, *tương tranh tài nguyên* là những thách thức trong cung cấp tài nguyên. Vì vậy, cần thiết phải có sự thiết lập trật tự hoặc chế độ ưu tiên và phương án dự phòng đối với tài nguyên.

1.1.2. Nguyên lý và giải pháp điều khiển cung cấp tài nguyên

Theo các tác giả Tanenbaum, Steen [124], Haddad và cộng sự [41] và Kshemkalyani, Singhal [60], bộ cung cấp tài nguyên là quá trình xử lý, tính toán để được cung cấp tối đa tài nguyên, tránh trường hợp tương tranh, bế tắc. Để nâng cao hiệu năng trong cung cấp tài nguyên, cần phải có sự phân tích, mô tả tài nguyên dựa trên các nguyên lý cung cấp tài nguyên. *Nguyên lý cung cấp tài nguyên* tập trung vào ba vấn đề cơ bản:

- *Phương pháp*: là đưa ra giải pháp cung cấp tài nguyên tối ưu cho các tiến trình yêu cầu từ các ứng dụng với chi phí tối thiểu. Hiệu năng của việc cung cấp là tài nguyên được sử dụng hiệu quả nhất trong quá trình cung cấp, không có trạng thái tương tranh và các tiến trình được đáp ứng đủ tài nguyên được yêu cầu.
- *Quy trình*: đó là các bước thực hiện nhằm thiết lập mục tiêu đối với tài nguyên.
- *Chiến lược cung cấp tài nguyên*: các chương trình ứng dụng phải xử lý, tính toán đủ tài nguyên để thực hiện cung cấp.

Như vậy, trước khi triển khai ứng dụng thì nhất thiết phải mô tả chi tiết tài nguyên trong pha thiết kế để xây dựng bộ điều khiển cung cấp tài nguyên.

Đối với *cung cấp tài nguyên dùng chung*, nguyên lý cung cấp phải phòng tránh để không làm giảm hiệu năng hoạt động của hệ do các hiện tượng "*sốc*" làm tăng các yêu cầu mà không được đáp ứng của một số tài nguyên. Hai chức năng cần thiết để đảm bảo cung cấp:

- *Thứ nhất*: phân tán các yêu cầu giữa các tài nguyên tương ứng có khả năng thỏa mãn, chức năng này gọi là phân phối tải. Trong hệ phân tán, đây là điều kiện để tránh tình trạng mà ở đó các yêu cầu đợi đến lượt được thỏa mãn trên một máy chủ bị đầy, trong khi đó các tài nguyên tương ứng lại rỗi rãi trên các máy chủ khác.
- *Thứ hai*: giới hạn số lượng các yêu cầu được phép cho một số tài nguyên. Việc giới hạn có thể thực hiện bằng cách hạn chế (tĩnh hay động) số lượng các tiến trình, số lượng các giao dịch được chọn sử dụng toàn bộ hay từng phần tài nguyên, trường hợp này là điều khiển tải tổng quát.

Hai chức năng này là điều kiện cần đối với bộ điều khiển cung cấp tài nguyên dùng chung. Bộ điều khiển cung cấp tài nguyên dùng chung phải đạt được các yêu cầu sau:

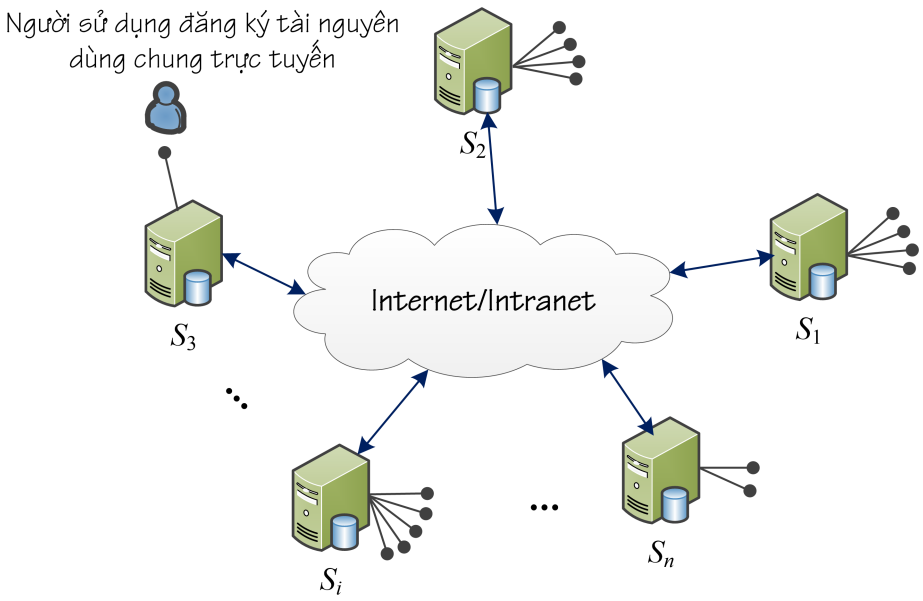
- Tuân thủ các quy tắc sử dụng tài nguyên nhằm tránh xảy ra tình trạng bế tắc và thiếu tài nguyên.
- Phân bố tải tương đối đồng đều giữa các tài nguyên cùng loại.
- Giới hạn nhu cầu nhằm duy trì hệ thống hoạt động đạt mức hiệu quả nhất định.

Trong Điện toán đám mây, các nghiên cứu trong [128, 112, 33, 84, 100, 117] trình bày nguyên lý cung cấp tài nguyên thực hiện các giải pháp tối ưu các cung cấp dịch vụ dựa trên nhu cầu. Bên cạnh đó, cung cấp tài nguyên và các thuật toán trong Điện toán đám mây sử dụng cho các yêu cầu và cấu hình khác nhau là rất khó để bảo toàn hiệu năng trong hệ thống đám mây. *Tóm lại, bộ điều khiển cung cấp tài nguyên trong Điện toán đám mây là áp dụng các thuật toán để tối ưu quá trình cung cấp đối với tài nguyên vật lý hoặc ảo hóa cho các ứng dụng nhằm nâng cao hiệu năng hoạt động của đám mây.*

Trong cung cấp tài nguyên của hệ tập trung chỉ xét kết nối giữa người sử dụng và các máy chủ theo truyền unicast nhưng trong cung cấp tài nguyên của hệ phân tán xét thêm phần truyền giữa các máy chủ với nhau theo cơ chế truyền multicast.

Hình 1.1 theo Chương II của tác giả Lê [65] mô tả tập các máy chủ kết nối qua môi trường truyền thông trong cung cấp tài nguyên dùng chung. Theo Hình 1.1, nếu

xét hoạt động mỗi máy chủ một cách độc lập, không có sự phối hợp để chia sẻ tài nguyên dùng chung thì đây là hệ tập trung. Nếu xét các máy chủ hợp lực để chia sẻ tài nguyên dùng chung thì đây là hệ phân tán. Sự hợp lực các máy chủ là sự phối hợp giữa các máy chủ với nhau để cung cấp tài nguyên dùng chung cho người sử dụng. Khác biệt giữa hệ tập trung và phân tán là các đặc tính như: tính gắn bó, khả năng chịu lỗi, sự mở rộng, cân bằng tải,... Các nghiên cứu trong [62, 105] trình bày về truyền thông trong phân tán đề cập đến cơ chế truyền multicast. Sự khác biệt cơ chế truyền multicast là gói tin vào/ra trong một máy chủ không tuân thủ nguyên tắc về lưu lượng như truyền unicast. Truyền multicast là sự kết hợp đặc biệt từ các máy chủ kết nối với máy chủ phát thông tin truyền.



Hình 1.1: Mô hình tổng quan kết nối trong hệ phân tán [65]

Nghiên cứu của Luận án trình bày các đặc điểm truyền multicast hữu ích cho việc xây dựng chiến lược cung cấp tài nguyên trong hệ thống phân tán như sau:

- *Dịch vụ chịu lỗi dựa trên bản sao (replication)*: dịch vụ bản sao bao gồm một nhóm các máy chủ kết nối thông qua môi trường truyền thông. Yêu cầu máy chủ là truyền thông điệp multicast cho tất cả các thành viên của nhóm, các thông điệp này thực thi một hoạt động như nhau. Khi một số thành viên bị lỗi, máy khách truy cập tài nguyên vẫn có thể được phục vụ.
- *Dịch vụ chuyển kết nối (migration)*: thông điệp multicast có thể được sử dụng bởi các máy chủ và máy khách để xác định vị trí dịch vụ để đăng ký tài nguyên

trong hệ thống phân tán. Khi một máy chủ bị sự cố hoặc quá tải, hệ thống sẽ chuyển đổi truy cập máy khách sang một máy chủ khác có khả năng xử lý cao hơn để đảm bảo quá trình đăng ký tài nguyên mà không ảnh hưởng đến quá trình xử lý.

- *Hiệu năng cao (High-performance) thông qua nhân bản dữ liệu*: Dữ liệu được nhân bản để tăng hiệu suất của việc cung cấp tài nguyên, trong một số trường hợp bản sao của dữ liệu được đặt trong các máy khách để việc đăng ký thuận tiện hơn. Mỗi lần thay đổi dữ liệu, giá trị mới được truyền multicast cho các tiến trình quản lý các bản sao.

- *Nhân bản các thông điệp*: truyền multicast cho một nhóm có thể được sử dụng để thông báo cho tiến trình khi diễn ra các hoạt động trên máy chủ. Truyền multicast quan trọng đối với song song hóa thuật toán Lamport [71] để cung cấp giá trị đồng hồ lô gíc, thiết lập trật tự tổng quát và thực hiện xử lý tuần tự trên các máy chủ.

Bên cạnh yếu tố truyền multicast trong hệ phân tán trong phạm vi phát triển của các loại ứng dụng khác nhau cùng tồn tại và phụ thuộc các giao thức trên Internet cho các ứng dụng. Ngoài ra, truyền thông IP được thực hiện trên một số lượng lớn kết nối và ngày càng tăng của công nghệ mạng. Theo báo cáo của tác giả Oanh [98] số lượng IP phiên bản 4 sắp cạn kiệt và địa chỉ IP multicast rất hạn chế. Các yếu tố trên đã dẫn đến sự quan tâm nghiên cứu *ảo hóa mạng*.

Theo các tác giả Chowdhury, Boutaba [18] và Haider, Potter, Nakao [42] trình bày *ảo hóa mạng* liên quan với việc xây dựng nhiều mạng ảo khác nhau trên một mạng hiện có như Internet. Mỗi mạng ảo có thể được thiết kế để hỗ trợ một ứng dụng phân tán cụ thể. Câu hỏi đặt ra là *ứng dụng cụ thể được xây dựng trên một mạng thực có thể thực thi trên một mạng ảo và tối ưu hóa cho ứng dụng này mà không thay đổi các đặc tính của mạng cơ bản?* Mạng thực đã có các chương trình xác định địa chỉ, giao thức và thuật toán định tuyến. Tương tự, mỗi mạng ảo cũng có thành phần trên nhưng xác định lại để đáp ứng nhu cầu các loại ứng dụng cụ thể. Xét mạng bao phủ (Overlapping Networks) trong cung cấp tài nguyên phân tán. Một mạng bao phủ là một mạng ảo bao gồm các nút và các liên kết ảo, nằm phần trên của một mạng cơ bản và cung cấp các đặc tính không có trong mạng cơ bản như:

- Các dịch vụ trong mạng bao phủ được thiết kế hướng nhu cầu của một loại ứng dụng cụ thể.

- Các hoạt động trong mạng bao phủ hiệu quả hơn trong một môi trường mạng nhất định.
- Mạng bao phủ có thê mở rộng thêm đặc tính như truyền multicast hoặc bảo mật,...

Dựa vào các đặc tính trên, việc xây dựng hệ phân tán trên mạng bao phủ có những ưu điểm:

- Cho phép xác định các dịch vụ mạng mới được xác định mà không cần thay đổi các dịch vụ mạng cơ bản. Một đặc điểm là mạng bao phủ đưa ra mức độ tiêu chuẩn hóa cũng như việc sửa đổi chức năng cơ bản trong các thiết bị định tuyến.
- Khuyến khích thử nghiệm với các dịch vụ mạng cải tiến, tự phát triển và các tuỳ biến dịch vụ thuộc ứng dụng cụ thể.
- Nhiều lớp bao phủ có thể được xác định và có thể cùng tồn tại. Kết quả cuối cùng là tạo ra kiến trúc mạng mới được mở rộng.

Ảo hóa mạng được triển khai trên các hệ thống ảo hóa để cung cấp tài nguyên truyền thông cho các máy chủ ảo. Kết quả của ảo hóa mạng là thiết lập các mạng ảo kết nối với nhau thông qua các bộ cung cấp và điều khiển. Để triển khai ảo hóa mạng nhất thiết xây dựng các đặc tính cho bộ cung cấp tài nguyên.

Tóm lại, giải pháp điều khiển cung cấp tài nguyên phân tán của Luận án tập trung vào cơ chế định tuyến multicast dựa trên các kênh truyền, tối ưu chi phí dựa vào đường dẫn để đạt được thông lượng cực đại đến tập đích dựa trên mạng ảo.

1.1.3. Bộ cung cấp tài nguyên

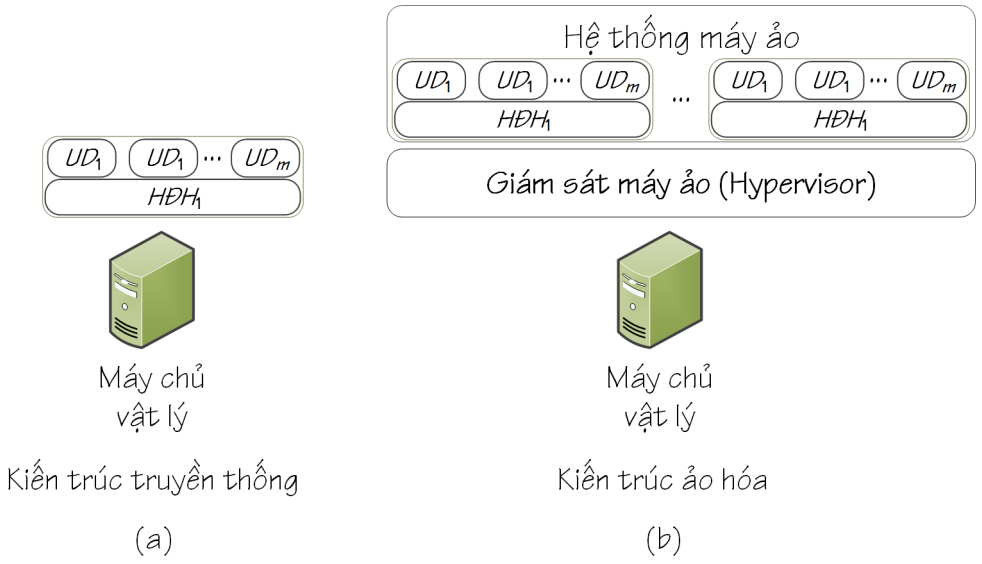
Theo các tác giả Tanenbaum, Steen [124], Haddad, Kordon, Petrucci [41] và Kshemkalyani, Singhal [60], cung cấp tài nguyên trong hệ phân tán là mô hình mở cho các bài toán tối đa hóa tiện ích cung cấp tài nguyên. Bộ cung cấp tài nguyên được xây dựng nhằm điều khiển, giám sát và cung cấp tài nguyên. Bộ cung cấp tài nguyên trong hệ phân tán RADS (Resource Allocator in Distributed System) hoạt động dựa trên các thuật toán cung cấp tất cả tài nguyên dùng chung sẵn có. Các tài nguyên trong hệ phân tán được cung cấp thông qua thông điệp yêu cầu từ người sử dụng. Bộ cung cấp đáp ứng tài nguyên sau khi ánh xạ được tài nguyên. Ngữ cảnh này có thể được nhận thấy thông qua các chương trình cung cấp tài nguyên dùng chung cho nhiều lĩnh vực như: thương mại điện tử, y tế, giáo dục, chính phủ điện tử,...

Nhược điểm của RADS là khi tài nguyên tới hạn đồng thời với số lượng yêu cầu tài nguyên lớn dẫn đến việc xử lý cung cấp tài nguyên chặt chẽ, nếu không có giải pháp tốt dễ dẫn đến tắc nghẽn, sự cố. Do đó, vấn đề điều khiển RADS lúc này cần thiết được xây dựng với giải pháp tối ưu. Theo các tác giả Lamport [71], Lê [65], Tanenbaum, Steen [124] và Kshemkalyani, Singhal [60], đối với cung cấp tài nguyên cho hệ phân tán đòi hỏi tài nguyên dùng chung phải được cấp phát duy nhất, do đó RADS cần phải có sự đồng bộ giữa các tiến trình trong hệ phân tán thông qua sự hợp lực giữa các máy chủ. Việc đồng bộ hóa tiến trình trong hệ phân tán dựa vào dấu, đó là giá trị đồng hồ lô gíc được đồng nhất trên tất cả các máy chủ. Để đạt được giá trị đồng hồ lô gíc, các thông điệp truyền đồng thời đến các máy chủ cùng lúc dựa vào truyền thông nhóm.

Theo Buyya, Broberg và Goscinski [9], dựa vào hệ thống ảo hóa theo Hình 1.2.b trong Phần 1.2.5, Phần I [9], các đám mây cung cấp dịch vụ cho người sử dụng thông qua các máy ảo. Vì vậy, cung cấp tài nguyên trên đám mây là quá trình cung cấp tài nguyên sẵn có một cách tối ưu và hiệu quả nhất cho các ứng dụng [112, 33, 84, 100, 117]. Theo các tác giả Sharkh và cộng sự [112] trình bày bộ điều khiển cung cấp tài nguyên nằm trong trong hệ thống giám sát máy ảo là quá trình lập lịch, ánh xạ các tài nguyên sẵn có với các hoạt động yêu cầu. Bộ cung cấp này phải giám sát, nhận biết và quản lý được tất cả các tài nguyên hiện có và yêu cầu từ người sử dụng. Bộ cung cấp tài nguyên dựa trên chiến lược quản lý tất cả hoạt động cung cấp với tài nguyên giới hạn trong môi trường đám mây để đáp ứng nhu cầu các ứng dụng trong đám mây. Công việc này đòi hỏi về loại và số lượng tài nguyên cần thiết cho mỗi ứng dụng để đáp ứng yêu cầu từ người sử dụng truy cập vào ứng dụng trên các máy ảo.

Bộ điều khiển cung cấp tài nguyên trong hệ thống máy chủ ảo có nhiệm vụ điều khiển quá trình chia sẻ tài nguyên qua Internet trong điều kiện tài nguyên có giới hạn. Các loại xử lý của bộ điều khiển cung cấp tài nguyên được xét:

- *Đặt trước*: tài nguyên đã được đặt trước và có cấu hình cụ thể, loại tài nguyên này nên sẵn có và dành riêng với thời gian cụ thể.
- *Tùy khả năng*: tài nguyên được cung cấp sớm nhất có thể, các yêu cầu tài nguyên này đang nằm trong hàng đợi chờ xử lý.
- *Tức thì*: khi có yêu cầu đến, tài nguyên được cung cấp ngay lập tức hoặc từ chối cung cấp dựa trên nguồn tài nguyên sẵn có.
- *Có thời hạn*: dựa vào quá trình lập lịch của bộ điều khiển, tài nguyên có thể được cung cấp và hoàn thành trước thời hạn đã đưa ra.



Hình 1.2: Kiến trúc truyền thống (a) và kiến trúc ảo hóa (b) [9]

Để đảm bảo tối ưu trong điều khiển cung cấp tài nguyên trong hệ thống máy chủ ảo, cung cấp tài nguyên truyền thông là một trong những yếu tố quan trọng khi thực hiện quá trình hợp lực giữa các máy chủ để cung cấp tài nguyên dùng chung. Bên cạnh đó, truyền thông đóng vai trò di trú của hệ thống máy chủ trong đám mây và đáp ứng dịch vụ đối với người sử dụng.

Bộ cung cấp tài nguyên truyền thông dựa trên ba nguyên lý cơ bản: đảm bảo kết nối, thông lượng cực đại và thời gian thực. Đối với nguyên lý thứ nhất, hệ thống truyền thông bao gồm nhiều kênh kết nối đến tập máy chủ, nếu một hay một số kênh bị sự cố không ảnh hưởng đến hoạt động của hệ. Nguyên lý thứ hai đạt được dựa vào các giải pháp và thuật toán trong truyền thông để đảm bảo rằng thông tin từ nguồn đạt được lưu lượng cực đại khi đến đích. Nguyên lý thứ ba là một trong những vấn đề cần được quan tâm tùy thuộc vào các ứng dụng khác nhau với yêu cầu thời gian thực khác nhau.

Tóm lại, bộ điều khiển cung cấp tài nguyên trong hệ thống máy chủ ảo theo nghiên cứu của Luận án là bộ điều khiển đảm bảo tối ưu quá trình cung cấp tài nguyên cho các ứng dụng hoạt động nhằm tăng hiệu năng của hệ thống đám mây.

1.2. Các nghiên cứu liên quan

1.2.1. Các nghiên cứu liên quan đến điều khiển cung cấp tài nguyên trong hệ phân tán

Hệ tập trung xuất hiện các hạn chế khi quy mô kết nối ngày càng tăng. Để đáp ứng sự phát triển đó, hệ phân tán được nghiên cứu và đưa vào triển khai ứng dụng. Các các vấn cần giải quyết của hệ phân tán so với hệ tập trung thể hiện qua Bảng 1.1.

Bảng 1.1: So sánh giữa hệ tập trung và hệ phân tán

Thành phần	**Hệ tập trung**	**Hệ phân tán**
Phần cứng	Khả năng mở rộng tài nguyên phần cứng rất khó khi tài nguyên tới hạn hoặc không còn khả năng cung cấp. Khi máy chủ gặp sự cố, hệ sẽ dừng hoạt động.	Dễ dàng thêm và mở rộng tài nguyên vật lý. Khi một máy chủ bị sự cố và rời khỏi hệ thống, hệ phân tán vẫn hoạt động bình thường. Hệ phân tán sẽ dừng hoạt động khi tất cả máy chủ bị sự cố đồng thời.
Phần mềm	Phần mềm được xây dựng độc lập cho máy chủ đáp ứng cho các ứng dụng đang truy cập, nếu vượt quá số lượng xử lý, hệ thống sẽ tạm dừng cung cấp tài nguyên dùng chung.	Hiệu năng cao hơn so với hệ tập trung, nếu một máy chủ bị sự cố thì các truy cập được chuyển cho máy chủ khác trong hệ đảm nhiệm xử lý tiếp công việc.
Dữ liệu	Cơ sở dữ liệu ở trạng thái cục bộ, khi xảy ra sự cố, hệ không đảm bảo tính gắn bó đối với người sử dụng. Khi bị mất hoặc hỏng dữ liệu, dữ liệu chỉ được thiết lập lại với các bản đã sao lưu trước đó.	Dữ liệu trên một máy chủ bị mất không làm mất dữ liệu cho toàn bộ hệ thống. Chia sẻ tài nguyên dùng chung: người sử dụng truy cập vào tài nguyên cục bộ trên chính máy chủ xử lý mà không cần biết đến tài nguyên ở các máy chủ khác trong hệ.
Truyền thông	Dễ xảy ra tắc nghẽn truyền thông khi số lượng truy cập vào hệ thống vượt nhu cầu cung cấp tài nguyên. Khi sự cố đường truyền vật lý, hệ sẽ không truy cập được.	Hệ thống tương tác người sử dụng chủ yếu thông qua truy cập tại máy chủ gần nhất dẫn đến tốc độ truy cập cao hơn.

Hệ phân tán bao gồm thành phần các máy chủ được kết nối với nhau và người sử dụng thông qua các ứng dụng. Cả hai thành phần này được gắn kết và hợp lực với nhau để cung cấp tài nguyên dùng chung cho người sử dụng. Theo Phần 1.3, Chương 1 của Tanenbaum và Steen trong [124], cung cấp tài nguyên người sử dụng trong hệ phân tán phân thành ba loại cơ bản: Hệ phân tán tính toán (High performance distributed computing), Hệ phân tán thông tin (Distributed information systems) và Hệ phân tán phức tạp (Distributed systems).

Hiện nay, các công trình nghiên cứu và công bố tài Việt Nam về lĩnh vực hệ phân tán còn hạn chế. Các công bố của tác giả Lê và cộng sự trình bày các giải pháp như:

đảm bảo tính gắn bó trong hệ phân tán [66, 68, 70], xử lý song song tiến trình trong môi trường phân tán [67], giám sát hệ phân tán [69, 127]. Đối với các nghiên cứu liên quan đến điều khiển cung cấp tài nguyên trong hệ phân tán, hai nhóm giải pháp tác giả Lê và cộng sự tập trung nghiên cứu là đảm bảo tính gắn bó và xử lý song song tiến trình trong hệ phân tán.

Dựa vào các phân loại hệ phân tán, Luận án tập trung vào *phát triển hệ phân tán phức tạp để cung cấp tài nguyên dùng chung*. Tuy nhiên, hệ phân tán có các hạn chế và thách thức trong nghiên cứu thể hiện trong Bảng 1.2. Thông qua Bảng 1.2, nhược điểm của hệ phân tán được liệt kê dựa trên bốn thành phần đặc trưng: phần cứng, phần mềm, dữ liệu và truyền thông. Dựa vào những nhược điểm này, Luận án tập trung giải quyết một số vấn đề còn tồn tại thuộc ba thành phần: *phần mềm, dữ liệu và truyền thông*.

Bảng 1.2: Ưu và nhược điểm của hệ phân tán

Thành phần	Ưu điểm	Nhược điểm
Phần cứng	Dễ dàng thêm và mở rộng tài nguyên vật lý. Khi một máy chủ bị sự cố và rời khỏi hệ thống, hệ phân tán vẫn hoạt động bình thường. Hệ phân tán sẽ dừng hoạt động khi tất cả máy chủ bị sự cố đồng thời.	Quá tải trên một máy chủ khi có số lượng lớn truy cập trong khi các máy chủ khác ít tải.
Phần mềm	Hiệu năng cao hơn so với hệ tập trung, nếu một máy chủ bị sự cố thì các truy cập được chuyển cho máy chủ khác trong hệ đảm nhiệm xử lý tiếp công việc.	Dễ xảy ra bế tắc trong quá trình cung cấp tài nguyên dùng chung. Vấn đề bảo mật của hệ thống khi chia sẻ tài nguyên dùng chung.
Dữ liệu	Dữ liệu trên một máy chủ bị mất không làm mất dữ liệu cho toàn bộ hệ thống. Chia sẻ tài nguyên dùng chung: người sử dụng truy cập vào tài nguyên cục bộ trên chính máy chủ xử lý mà không cần biết đến tài nguyên ở các máy chủ khác trong hệ.	Quản trị hệ thống và cơ sở dữ liệu trở nên khó khăn khi các máy chủ phân tán. Các gói tin bị trùng lắp khi truyền theo phương thức multicast.
Truyền thông	Hệ thống tương tác người sử dụng chủ yếu thông qua truy cập tại máy chủ gần nhất dẫn đến tốc độ truy cập cao hơn.	Hợp lực các máy chủ thông qua cơ chế truyền thông điệp dẫn đến tiêu tốn tài nguyên truyền thông. Băng thông không đáp ứng đủ khi truyền với dữ liệu lớn. Thất lạc gói tin trong quá trình truyền.

Hai hướng tiếp cận nghiên cứu trong hệ phân tán đó là giải pháp đảm bảo gắn bó đối với người sử dụng (đối với thành phần phần mềm, dữ liệu) và tối ưu truyền thông điệp giữa các máy chủ (đối với thành phần truyền thông). Để giải quyết các hướng

nghiên cứu, *Luận án đã tìm hiểu và phân tích để áp dụng, cải tiến và phát triển các thuật toán truyền thông và điều khiển thông điệp.*

1.2.1.1. Các thuật toán truyền thông trong hệ phân tán

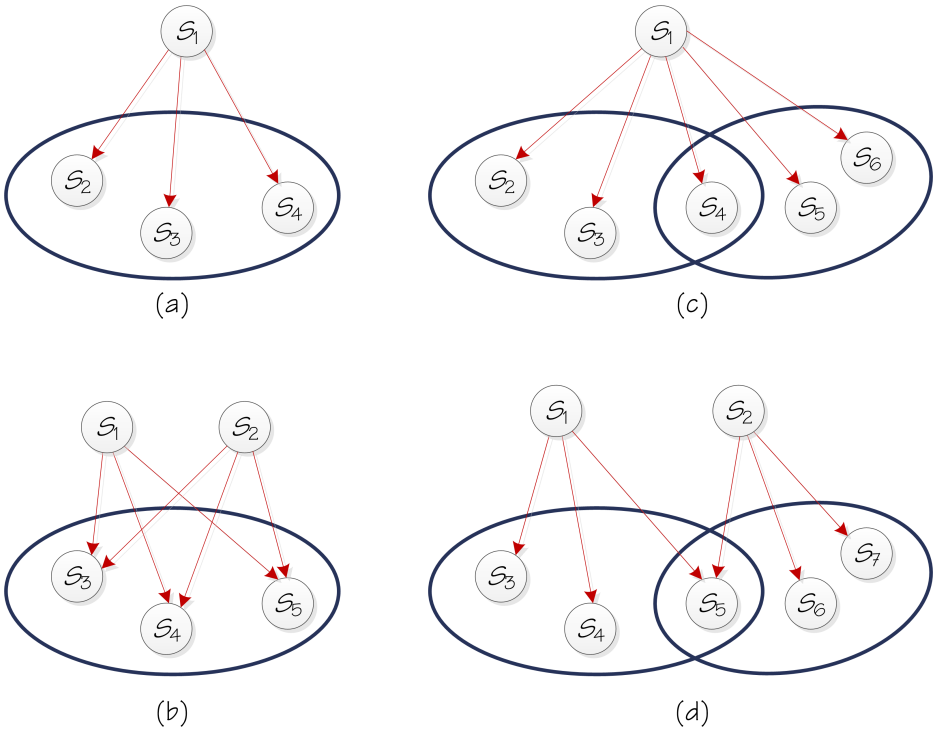
Trong các thiết bị phần cứng hiện nay, hỗ trợ truyền broadcast và multicast được cung cấp bởi các ngăn xếp giao thức mạng sử dụng các biến của cây bao trùm. Dó là một cơ chế hiệu quả cho việc phân tán thông tin. Tuy nhiên, hỗ trợ phần cứng hoặc hỗ trợ giao thức tầng mạng của IP multicast có thể không có hiệu quả khi gặp các đặc tính sau:

- Vấn đề trật tự của ứng dụng đặc trưng dựa trên trật tự cung cấp các thông điệp.
- Thích ứng với các nhóm động thay đổi thành viên.
- Gửi multicast đến một tập bất kỳ của các tiến trình tại mỗi sự kiện gửi.
- Các vấn đề trong cung cấp tài nguyên thay đổi.
- Khả năng chịu lỗi.

Để giải quyết hạn chế các đặc tính vừa nêu, cải tiến phương thức truyền IP multicast sang truyền multicast. Nếu thuật toán truyền multicast yêu cầu bộ gửi là một phần của nhóm đích, thuật toán truyền multicast được gọi là thuật toán nhóm kín. Nếu thuật toán truyền multicast đến nhóm nằm ngoài nhóm đích, thuật toán truyền multicast được gọi là thuật toán nhóm mở. Các thuật toán nhóm mở tổng quát hơn và do đó khó hơn về giải thuật để thiết kế và chi phí cao hơn để thực hiện so với nhóm kín. Các thuật toán nhóm kín được sử dụng trong một số ngữ cảnh như: đăng ký tài nguyên trực tuyến, các hệ thống ngân hàng điện tử,... nơi tiến trình máy khách có thời gian thực thi ngắn và số lượng truy cập lớn.

Các loại thuật toán truyền multicast trong Phần 6.9 của các tác giả Kshemkalyani và Singhal [62] trình bày bốn lớp của các mối quan hệ truyền thông từ nguồn đến tập đích, tô pô theo Hình 1.3 thể hiện lớp các nhóm mở:

- Hình 1.3(a): nhóm đơn nguồn và đơn đích.
- Hình 1.3(b): nhóm đa nguồn và đơn đích.
- Hình 1.3(c): các nhóm đơn nguồn và đa đích, có thể chồng chéo.
- Hình 1.3(d): các nhóm đa nguồn và đa đích, có thể chồng chéo.



Hình 1.3: Tô pô các loại thuật toán truyền multicast [62]

Các nhóm theo Hình 1.3(a) và Hình 1.3(c) là đơn giản để thực hiện, giả sử sự hiện diện của các kênh FIFO giữa mỗi cặp của các tiến trình thì cả hai trật tự toàn phần và trật tự nhân quả được đảm bảo. Nhóm theo Hình 1.3(b) độ phức tạp để xử lý chưa cao, nhưng sự chuyển đổi lớp này đến nhóm theo Hình 1.3(d) thì khá phức tạp. Đối với phương pháp thiết kế cho các nhóm theo Hình 1.3(d), cách tiếp cận này thường được gọi là tiếp cận cây lan truyền, sử dụng một cấu trúc bán tập trung có các thuật toán thích nghi tập trung và được đề xuất bởi các tác giả Chiu, Hsaio [14] và Jia, Zhou [52].

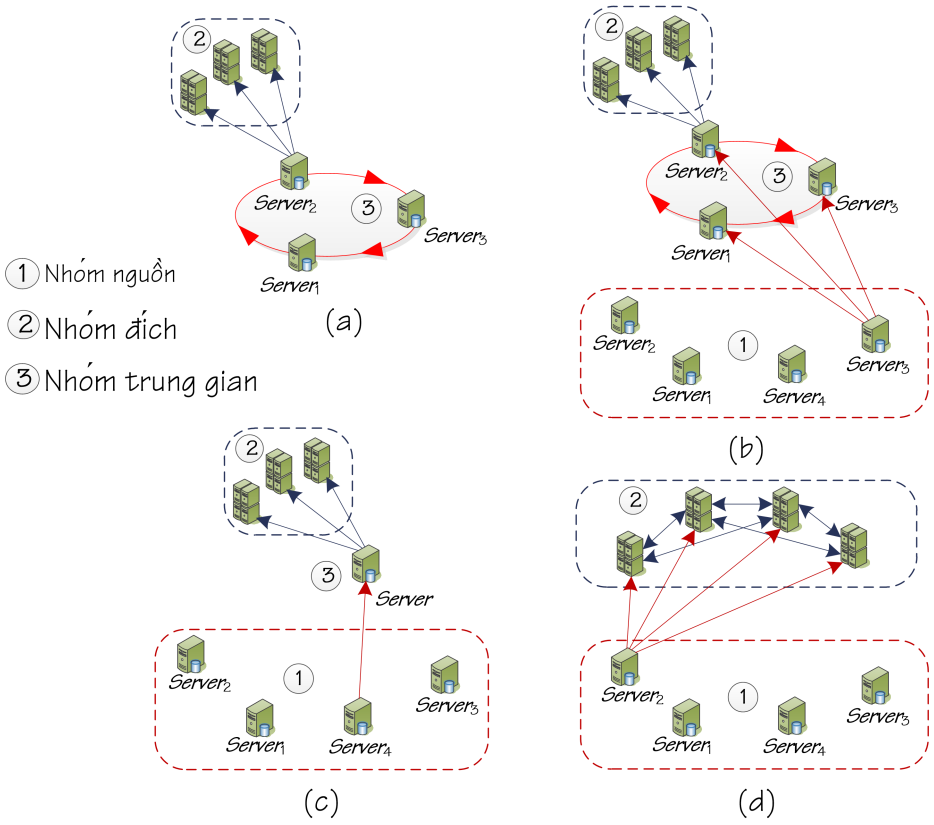
Các thuật toán truyền multicast mô tả đặc tính kỹ thuật thách thức trong việc thiết kế tô pô. Kịch bản tổng quát nhất cho phép mỗi tiến trình truyền multicast trong tô pô đến một nhóm tùy ý và nhóm thay đổi động của tiến trình ở mỗi bước. Kịch bản tổng quát phát sinh chi phí nhiều hơn, các thuật toán thực hiện trên hệ thống thực thường là "*tập trung*" hơn theo một hướng. Nhiều phương thức truyền multicast đã được phát triển và triển khai, nhưng tất cả đều có thể được phân loại thuộc về các lớp sau:

Thuật toán truyền thông dựa vào lịch sử (Communication history-based)

Các nghiên cứu về thuật toán trong lớp này sử dụng một phần của lịch sử truyền

thông để đảm bảo trật tự các yêu cầu. Nghiên cứu thuật toán cơ bản trật tự nhân quả phân tán được trình bày trong Phần 12.2 trong [104] theo đề xuất bởi Raynal, Schiper, Toueg và thuật toán tối ưu đối với trật tự nhân quả tổng quát trong [60, 61] đề xuất bởi Kshemkalyani và Singhal chỉ ra giải quyết vấn đề cung cấp trật tự nhân quả trong triển khai phân tán. Các thuật toán không cần lưu vết các nhóm riêng và do đó làm việc trên truyền multicast nhóm mở.

Thuật toán Lamport được sử dụng để gán nhãn thời gian cho các thông điệp được đề xuất bởi Lamport trong [71]. Thông điệp được gán nhãn thời gian chỉ khi nó biết rằng không có thông điệp khác với nhãn thời gian thấp hơn được gán và truyền multicast. Giao thức NewTop đề xuất bởi Ezhilchelvan, Macdo và Shrivastava trong [32] được mở rộng từ thuật toán Lamport cho các nhóm chồng chéo cũng đảm bảo cả trật tự toàn phần và nhân quả. Các thuật toán triển khai trên giao thức NewTop sử dụng cho cấu hình nhóm đóng.



Hình 1.4: Mô hình tuần tự thông điệp [104]

Thuật toán dựa trên quyền (Privilege-based)

Các nghiên cứu về thuật toán dựa trên quyền được thực thi theo mô tả trong

Hình 1.4.a. Một thẻ bài (token) tuần hoàn giữa các tiến trình gửi, thẻ bài mang số tuần tự cho thông điệp tiếp theo được truyền multicast và chỉ có máy chủ giữ thẻ có thể truyền multicast. Sau khi một sự kiện gửi multicast, số tuần tự được cập nhật, các tiến trình đích cung cấp thông điệp trong trật tự tăng dần số tuần tự. Bộ gửi và nhận cần phải nhận biết lẫn nhau, do đó được giả định cho các nhóm kín. Như vậy thuật toán có thể cung cấp trật tự toàn phần, cũng như trật tự nhân quả sử dụng cấu hình nhóm kín.

Thuật toán di chuyển tuần tự (Moving sequencer)

Các nghiên cứu về thuật toán di chuyển tuần tự được thực thi theo minh họa trong Hình 1.4.b. Các tác giả Chang và Maxemchuck trình thuật toán nguyên thủy trong [12] và phiên bản khác nhau của nó đã được đưa ra bởi các thuật toán Pinwheel và RMP. Các bước thuật toán đã được đề xuất thực hiện theo Bảng 1.3.

Bảng 1.3: Các bước thực hiện thuật toán di chuyển tuần tự

Các bước	**Thực hiện**
Bước 1	Để truyền multicast một thông điệp, bộ gửi gửi thông điệp đến tất cả các bộ tuần tự.
Bước 2	Bộ tuần tự lưu hành một thẻ bài với nhau. Thẻ bài mang một số tuần tự và một danh sách của tất cả các thông điệp mà số thứ tự đã được gán, đó là các thông điệp đã được gửi đi rồi.
Bước 3	Khi bộ tuần tự nhận được thẻ bài, nó gán số tuần tự cho tất cả các thông điệp đã nhận nhưng không theo thứ tự. Sau đó nó sẽ gửi các thông điệp tuần tự mới đến các đích, chèn các thông điệp vào danh sách thẻ bài, và chuyển các thẻ bài đến bộ tuần tự kế tiếp;
Bước 4	Các tiến trình đích cung cấp thông điệp đã nhận trong trật tự tăng dần số tuần tự.

Trong nghiên cứu thuật toán di chuyển tuần tự của tác giả Lê đề xuất trong [65]; xét một mạng kết nối với nhau qua giao thức TCP/IP là một tập hợp bao gồm N máy chủ, trong đó một máy chủ có thể liên lạc với các máy chủ khác một cách dễ dàng. Một số duy nhất bao gồm từ 0 đến n-1 được phân phối một lần cho toàn bộ trên từng máy chủ. Máy chủ S_i đều có máy chủ hàng xóm bên phải hay còn gọi máy chủ kế tiếp sau mà số của máy chủ đó là $suc\,[S_i]$ và hàng xóm bên trái hay còn gọi là máy chủ liền kề trước mà số của nó là $pred\,[S_i]$. Sự mô tả này được gọi là một vòng tròn ảo. Khi hoạt động bình thường, n máy chủ được thể hiện đầy đủ trên vòng tròn ảo theo Công thức 1.1:

$$suc\,[S_i] = S_i + 1 \text{ modulo } n \tag{1.1}$$

$$pred\,[S_i] = S_i \text{ - } 1 \text{ modulo } n$$

Thuật toán trình bày dựa trên nguyên tắc như sau: ấn phong được cụ thể hóa trên một vài cấu hình của các biến trạng thái và di chuyển trên vòng tròn ảo theo một chiều xác định. Để vòng tròn ảo có thể hoạt động tốt, thì cần thiết phải xây dựng lại

vòng tròn ảo khi có một máy chủ nào đó bị sự cố. Phép xử lý này gọi là cấu hình lại bao gồm cả việc cập nhật các giá trị của $suc[i]$ và $pred[i]$ của các máy chủ hàng xóm bị sự cố. Việc cấu hình lại mạng được thực hiện nếu máy chủ bị sự cố không phá huỷ liên kết lô gíc. Khi một máy chủ bị sự cố có thể hoạt động trở lại thì phép toán chèn cho phép nó tham gia vào mạng.

Thuật toán di chuyển tuần tự đảm bảo trật tự toàn phần

Thuật toán bộ tuần tự cố định (Fixed sequencer)

Các nghiên cứu về thuật toán bộ tuần tự cố định được thực thi theo minh họa trong Hình 1.4.c. Phương pháp tiếp cận nghiên cứu cây lan truyền thuộc về lớp thuật toán này. Các thuật toán khác là bộ tuần tự tổng quát Isis do Birman và Joseph đề xuất trong [4], các đặc tính truyền thông nhóm, thảo luận và đánh giá về hệ thống Amoeba, Phoenix, Newtop được trình bảy bởi các tác giả Defago, Schiper, Urban trong [25], Chockler và cộng sự trong [15].

Tất cả tiến trình duy trì đồng hồ lô gíc và mỗi nhóm đều có một bộ tuần tự độc lập. Một tiến trình thuộc về nhiều nhóm cần phải gửi thông điệp tiếp theo (đến bộ tuần tự có liên quan) cho đến khi nó đã nhận và xử lý tất cả các thông điệp từ bộ tuần tự khác nhau; tương ứng với các thông điệp trước đó đã gửi. Hàng đợi FIFO phải duy trì trật tự tổng quát.

Thuật toán chấp nhận đích (Destination agreement)

Nghiên cứu thực thi thuật toán chấp nhận đích theo minh họa trong Hình 1.4.d. Trong lớp của thuật toán này, những đích nhận các thông điệp với một số thông tin trật tự hạn chế. Sau đó các máy chủ trao đổi thông tin với nhau để xác định một trật tự. Có hai lớp con: 1. sử dụng nhãn thời gian, 2. sử dụng chấp nhận hay giao thức "*đồng thuận*" giữa các tiến trình.

Để triển khai các hệ phân tán dựa trên truyền multicast, những nghiên cứu trong hệ thống ảo hóa đề xuất ảo hóa mạng để đáp ứng cơ chế và các thuật toán truyền thông. Nhưng triển khai ảo hóa mạng sẽ mạng lại nhiều vấn đề phức tạp trong quá trình tối ưu tài nguyên truyền thông.

Tóm lại, các thuật toán truyền thông cho hệ phân tán đáp ứng cho các ứng dụng triển khai. Tùy thuộc vào từng tô pô và ứng dụng cụ thể để áp dụng các thuật toán khác nhau. Trên cơ sở nghiên cứu phương thức truyền multicast, *Luận án tập trung vào phát triển các thuật toán theo Hình 1.3.c; đó là, thuật toán dành cho nhóm kín, dựa trên quyền và di chuyển tuần tự đảm bảo trật tự toàn phần.*

Đối với thuật toán dành cho *nhóm kín* và *đảm bảo trật tự toàn phần*, Luận án đề xuất song song hóa thuật toán Lamport trình bày Mục 2.1 trong Chương 2. Đối với thuật toán *dựa trên quyền* và *di chuyển tuần tự đảm bảo trật tự toàn phần* Luận án đề xuất thuật toán 4PCoDT trình bày Mục 2.2 trong Chương 2.

Trong các nghiên cứu của các tác giả Raynal, Tel, Chockler và cộng sự trình bày trong [31, 105, 126] cũng chỉ ra một phần tối ưu truyền thông dựa vào các thuật toán xử lý trên cây trong truyền multicast. Tuy nhiên, hạn chế của cơ chế truyền này là vấn đề dư thừa trong truyền thông nếu nhiều máy chủ cùng cung cấp thông tin đến các máy chủ đích hoặc máy khách [72, 53, 16]. Để khắc phục nhược điểm trên, Luận án đưa ra giải pháp loại bỏ các gói tin dư thừa nếu máy chủ hoặc máy khách đã nhận được, đó là *kỹ thuật mã mạng*.

1.2.1.2. Vấn đề kỹ thuật mã mạng điều khiển cung cấp tài nguyên truyền thông

Các tác giả Tan và cộng sự trong [123], Sattari, Fragouli, Markopoulou trong [110], Wu, Chiang, Kung trong [135] trình bày giải pháp điểu khiển tỷ lệ nguồn với mã mạng kết hợp cơ chế truyền multicast và thuật toán điều khiển tỷ lệ thích nghi với đồ thị con kết hợp với mã mạng. Tác giả Tan và cộng sự trong [123] đã chỉ ra ưu điểm cây multicast kết hợp mã mạng đạt hiệu quả tốt hơn về tỷ lệ truyền đối với cây multicast không có mã mạng. Tác giả Tan và cộng sự sử dụng giải thuật di truyền để tối ưu thông lượng và phòng tránh tắc nghẽn; bên cạnh đó, đối với các gói tin truyền multicast, giải thuật di truyền cải tiến có thể tìm kiếm và xây dựng cây multicast sử dụng tối thiểu các nút trung gian để tăng thông lượng. Hướng nghiên cứu của tác giả Tan và cộng sự xây dựng cấu trúc cây sẽ xét trường hợp lưu lượng gói tin cho bài toán đa nguồn đến đa đích.

Các tác giả Chou, Wu, Jain trong [17], Deb, Srikant trong [24] và Sarkar, Tassiulas trong [109] trình bày các giải pháp tập trung vào điều khiển tỷ lệ nguồn và lưu lượng truyền trong mạng nhằm chống tắc nghẽn trong quá trình truyền, đảm bảo trạng thái cân bằng trong lưu lượng.

Các tác giả Xi, Yeh [137], Wu, Chiang, Kung [135] và Khreishah, Khalil, Wu [56] trình bày các giải pháp trong mã mạng tối ưu truyền multicast trong hệ phân tán. Trong nghiên cứu mã mạng đưa ra hai vấn đề nghiên cứu truyền multicast đó là truyền phi tuyến [80] và truyền tuyến tính [47, 78] áp dụng cho hai môi trường truyền thông là có dây và không dây. *Nghiên cứu của Luận án chỉ xét trường hợp truyền tuyến tính cho các mạng có dây.*

Trong quá trình truyền thông các máy ảo, việc tối ưu các tỷ lệ truyền gói tin cần quan tâm nghiên cứu. Khi tỷ lệ truyền gói tin đảm bảo việc truyền thông giữa các máy ảo sẽ mang lại lợi ích, hiệu quả lớn trong các ứng dụng phân tán. Hệ phân tán triển khai trên máy ảo xét truyền thông như là một phần tài nguyên để hoạt động. Do đó, tài nguyên truyền thông sẵn có được tối ưu thì hệ phân tán trao đổi thông điệp càng hiệu quả.

Tác giả Zhang và cộng sự [143] trình bày ảo hóa chức năng mã mạng trên Internet. Các tác giả nhấn mạnh bộ cung cấp tài nguyên phải giải quyết tốt hai vấn đề cung cấp tài nguyên tính toán và tài nguyên mạng. Dựa vào ảo hóa chức năng mạng, các tác giả đề xuất thực thi mã mạng như chức năng mạng mới và triển khai trong đám mây phân tán. Kết quả nghiên cứu của các tác giả nêu ra hiệu quả của mã mạng trong truyền multicast tốt hơn so với các giải pháp heuristic.

Các nghiên cứu về mã mạng trong các máy ảo tập trung vào việc di trú, lưu trữ phân tán và bảo mật được trình bày trong [142, 6, 81]. Các nghiên cứu này cũng chỉ ra việc áp dụng kỹ thuật mã mạng đem lại nhiều hiệu quả trong truyền thông, đối với trường di trú khi các máy ảo cần tài nguyên và di trú sang hệ thống ảo khác thì nhu cầu băng thông lúc này đạt mức cực đại có thể giúp cho việc di trú nhanh nhất; trường hợp lưu trữ phân tán cũng mang tính chất tương tự đối với tập tin dung lượng lớn; đối với trường hợp bảo mật, đây là sự kết hợp giao thức bảo mật mã mạng và giao thức bảo mật lưu trữ đám mây. Ngoài ra, mã mạng được nghiên cứu trong trường hợp các đám mây di động được đề xuất bởi Fitzek và cộng sự [35].

Hướng nghiên cứu của Luận án đưa ra mô hình tối ưu hóa và đề xuất các thuật toán thích nghi, điều khiển tỷ lệ phân tán cho mã mạng dựa trên các dòng multicast. Bên cạnh đó, giải pháp tối ưu truyền thông multicast với mã mạng là thông tin tại tập đích tránh trùng lặp và đạt thông lượng tối ưu. Giải pháp được thực hiện trên cây multicast là xây dựng lại tô pô kết hợp kỹ thuật mã mạng để thông lượng đạt cực đại tại tập đích. Các thuật toán trong nghiên cứu sẽ được thực nghiệm, đánh giá và có thể được mở rộng để điều khiển truyền thông điệp trong nhiều ứng dụng khác nhau.

1.2.1.3. Vấn đề kỹ thuật trong nhãn thời gian lô gíc

Dựa trên phân loại hệ phân tán trình bày trong Mục 1.2.1 và các thuật toán truyền thông hệ phân tán trình bày trong Mục 1.2.2. Hệ phân tán hoạt động dựa trên *thuật toán nhóm kín* và *đảm bảo trật tự toàn phần hay còn gọi là trật tự tổng quát chặt chẽ*. Trong *thuật toán nhóm kín*, hệ phân tán không có đồng hồ vật lý toàn cục để gán giá trị cho các tiến trình. Do đó, hệ phải sử dụng đồng hồ riêng là *đồng hồ lô gíc* hay còn

gọi là *nhãn thời gian* để gắn giá trị và đồng bộ tiến trình trên các máy chủ được gọi là *nhãn thời gian lô gíc*.

Nhãn thời gian lô gíc được xây dựng dựa trên thuật toán Lamport trình bày trong [82, 65, 124]. Thuật toán Lamport nhằm cho phép ghi lại các sự kiện của hệ phân tán. Thuật toán tập trung vào nguyên lý sau: mỗi máy chủ S đều có trang bị công tơ với các giá trị nguyên gọi là H_{S_i}. Đó chính là đồng hồ lô gíc tăng lên giữa hai sự kiện kế tiếp. Máy chủ e phát thông điệp ghi dấu E của mình dựa trên giá trị hiện hành của H_{S_e}. Khi nhận được thông điệp, máy chủ nhận r cập nhật đồng hồ H_{S_r} riêng của mình bằng giải thuật rút gọn theo Công thức (1.2):

$$\begin{aligned} &If\ \ ((H_{S_r}) \leq E)\ \ then \\ &\qquad H_{S_r} := E + 1 \\ &\qquad EndIf \end{aligned} \tag{1.2}$$

Một sự kiện a (sk_a) sinh ra trong máy chủ i (S_i) và được đánh dấu bởi đồng hồ cục bộ gọi là $H_{S_i}(a)$. Nếu sk_a và sk_b đều là hai sự kiện gửi từ máy chủ S_i đến S_j, ta luôn luôn có quan hệ xác định theo Công thức (1.3) như sau:

$$sk_a \rightarrow sk_b \Leftrightarrow H_{S_i}(a) < H_{S_j}(b) \tag{1.3}$$

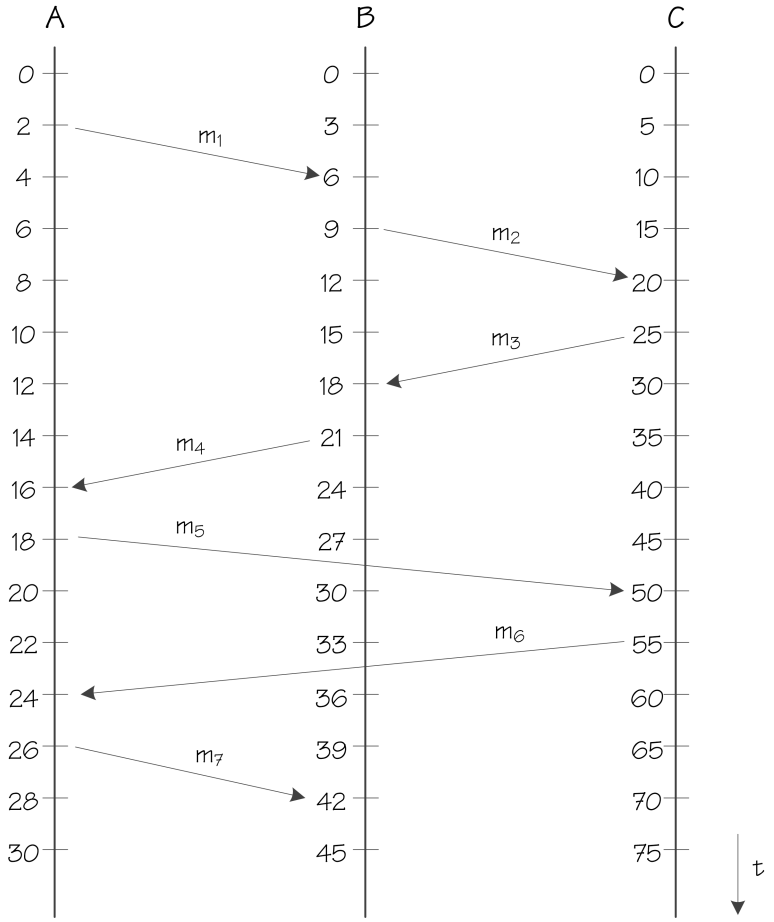
Trong đó: $sk_a \rightarrow sk_b$ thể hiện sự kiện a gửi cho sự kiện b. $\Leftrightarrow$ biểu thị cho phép kéo theo. $H_{S_i}(a) < H_{S_j}(b)$ thể hiện giá trị đồng hồ máy chủ a nhỏ hơn giá trị đồng hồ cục bộ máy chủ b.

Tuy nhiên, các nhãn đồng hồ này phải được cập nhật và nhất quán trên tất cả các máy chủ. Nếu giá trị không được cập nhật thì việc xử lý thông điệp sẽ bị sai và hoạt động của hệ sai trật tự theo lý thuyết trật tự như Hình 1.5 được trích dẫn trong Chương IV của tài liệu [65].

Theo Hình 1.5, khi các thông điệp di chuyển qua các máy chủ, giá trị gửi và nhận có giá trị khác nhau. Chính vì giá trị đồng hồ sai lệch, khi một máy chủ phát lệnh xử lý đồng thời trên các máy chủ sẽ dẫn đến sai lệch về các tiến trình được triệu gọi để xử lý. Do đó, *dữ liệu không nhất quán trên tất cả các máy chủ.*

Các máy chủ hoạt động nhận và gửi thông điệp dựa trên đồng hồ cục bộ của mình theo cơ chế truyền unicast, các máy chủ chỉ biết được *trật tự từng phần* trên máy chủ của mình và không nhận biết được các hoạt động trên máy chủ khác. *Trật tự từng phần* ảnh hưởng đến hoạt động tổng quát trong hệ phân tán, hai vấn đề cơ bản bị tác động đó là: 1. Giá trị đồng hồ lô gic trên các máy chủ không nhất quán; 2. Tiến

trình yêu cầu vào miền găng phải chờ đợi cho đến khi nhận đủ thông điệp có thể gây ảnh hưởng đến các máy chủ khác hoặc sai lệch khi tiến hành cập nhật dữ liệu. Để giải quyết bài toán trật tự từng phần, Luận án đã áp dụng và *song song hóa thuật toán Lamport* để xây dựng *trật tự tổng quát chặt chẽ* trên các máy chủ được trình bày trong Mục 2.1.1.

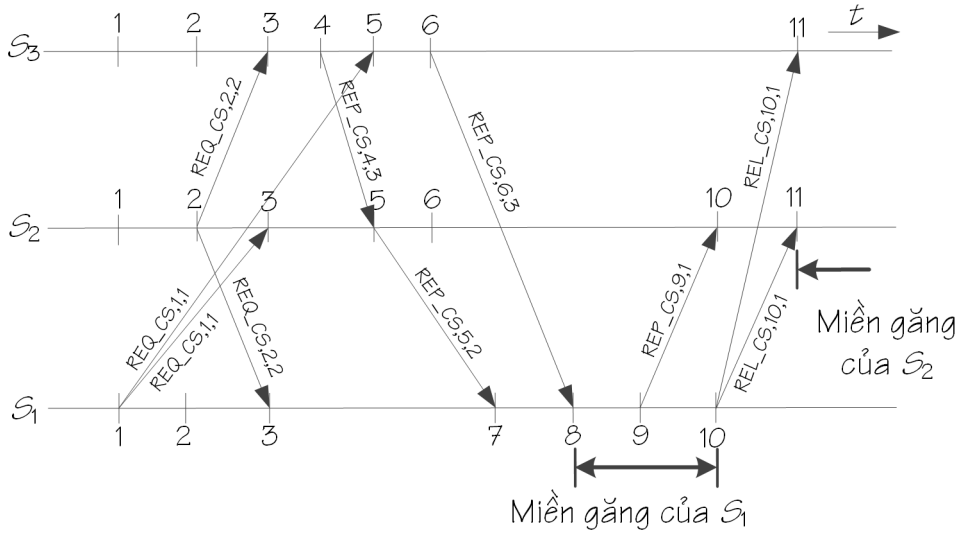


Hình 1.5: Nhãn thời gian thông điệp không theo trật tự

Trường hợp ví dụ cho thuật toán Lamport được trích từ Mục IV.2 của tác giả Lê trong [65]. Hình 1.6 thể hiện loại trừ tương hỗ nhờ dấu mô tả tiến trình vào miền găng của máy chủ 1 và máy chủ 2.

Thuật toán loại trừ tương hổ có thể mô tả như sau:

- Máy chủ S_i của mạng có thể gửi cho các máy chủ khác thông điệp có dạng *(T, H_{S_i}, S_i)*, trong đó H_{S_i} là dấu của thông điệp có nghĩa là đồng hồ lô gíc của nó và T có thể nhận một trong ba giá trị *REQ-CS (yêu cầu vào miền găng)*, *REP-CS*



Hình 1.6: Loại trừ tương hỗ nhờ dấu [65]

(phản hồi chấp nhận yêu cầu vào miền găng) và *REL-CS (giải phóng, rời khỏi miền găng)*. Thông điệp *REQ-CS* được phát đi cho tất cả các máy chủ, khi máy chủ S_i muốn vào trong miền găng. Thông điệp *REP-CS* được gửi bởi máy chủ S_j cho máy chủ S_i sau khi nhận được thông điệp *REQ-CS*. Thông điệp *REL-CS* được phát đi cho tất cả các máy chủ, khi máy chủ S_i đã rời khỏi miền găng.

- Mỗi máy chủ quản lý một hàng đợi các thông điệp được sắp xếp hoàn toàn bởi quan hệ $\Rightarrow$ theo cặp <thời gian, số> của từng thông điệp. Theo cấu trúc, hàng đợi luôn luôn chứa một thông điệp và chỉ một thường trực trong mỗi máy chủ, bao gồm máy chủ cục bộ. Khi có một thông điệp được gửi đi bởi máy chủ S_i, đồng thời nó cũng được ghi trong hàng đợi của máy chủ này.

- Trên mỗi máy chủ, khi nhận được một thông điệp dạng *(REQ-CS,* H_{S_i}*,* S_i*)* hay *(REL-CS,* H_{S_i}*,* S_i*)*, thông điệp này thay thế thông điệp M_i bất chấp nó là gì. Khi nhận thông điệp loại *(REP-CS,* H_{S_i}*,* S_i*)*, thông điệp này thay thế *Mi* ngoại trừ nếu M_i là một yêu cầu mà trong trường hợp đó *REP-CS* bị bỏ qua. Do vậy, ta có thể tiết kiệm việc gửi đi thông điệp *REP-CS* cho máy chủ S_i khi máy chủ này đã gửi một thông điệp *REQ-CS* và không còn thông điệp *REL-CS*.

- Máy chủ S_i được quyền vào miền găng khi thông điệp *REQ-CS* của nó đến trước theo nghĩa của quan hệ $\Rightarrow$ tất cả các thông điệp khác trong hàng đợi của nó.

Nguyên lý vào miền găng được phát biểu như sau: *Khi máy chủ* S_i *vào miền găng có nghĩa là nó đã nhận thông điệp từ tất cả các máy chủ khác và tất cả các thông điệp*

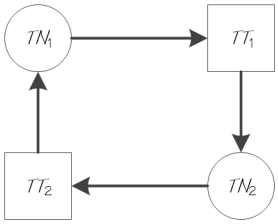
đều sau thông điệp REQ-CS của riêng nó.

Theo lý thuyết trật tự từng phần, hoạt động diễn ra trên các máy chủ trong Hình 1.6 thể hiện qua Bảng 1.4. Nếu dựa vào hoạt động trật tự từng phần, thì trên mỗi máy chủ chỉ biết các hoạt động đang diễn ra tại chính máy chủ của mình mà không không nhận biết được các hoạt động trên các máy chủ khác. Ví dụ tại máy chủ S_1 có các sự kiện chỉ diễn ra và nhận biết được tại các thời điểm giá trị đồng hồ lô gic: 1, 3, 7, 8, 9, 10. Như vậy, theo nguyên lý vào miền găng, máy chủ S_1 vào miền găng tại giá trị đồng hồ là 8.

Thông qua Hình 1.6 và Bảng 1.4, hai máy chủ S_1 và S_2 phát đi yêu cầu để vào miền găng. Trong trường hợp các yêu cầu này cùng truy cập vào một tài nguyên dùng chung dẫn đến tình trạng tương tranh tài nguyên. Trong quá trình hợp lực, nếu nhiều tiến trình cùng tương tranh tài nguyên dùng chung dễ dẫn đến bế tắc trong cung cấp tài nguyên.

Theo nghiên cứu của Singhal trong [115], quá trình bế tắc diễn ra khi hai hay nhiều tiến trình chiếm giữ tài nguyên dùng chung được giới hạn và đồng thời tiếp tục phát đi yêu cầu tài nguyên đang bị chiếm giữ tài nguyên khác. Các quá trình này tạo ra một vòng tròn khép kín làm cho các tiến trình kẹt chéo lẫn nhau dẫn đến bế tắc trong cung cấp tài nguyên theo mô tả trong Hình 1.7.

Nhiều giải pháp tính toán phân tán liên quan đến việc chia sẻ tài nguyên dùng chung giữa các tiến trình khác nhau đòi hỏi tài nguyên phải được cung cấp duy nhất cho một tiến trình tại một thời điểm. Do đó, loại trừ tương hỗ là giải pháp trong các hệ phân tán cung cấp tài nguyên dùng chung. Giải pháp loại trừ tương hỗ được giải quyết dựa trên đồng bộ hóa tiến trình truy cập vào các tài nguyên dùng chung để đảm bảo tính nhất quán và gắn bó trong hệ phân tán. Quá trình đồng bộ hóa bằng cách truyền thông điệp giữa các máy chủ dựa vào môi trường truyền thông. Loại trừ tương hỗ phân tán tuân thủ các yêu cầu sau:



Hình 1.7: Sự kẹt chéo của các tiến trình gây bế tắc [115]

Bảng 1.4: Hoạt động diễn ra trên các máy chủ trong trật tự từng phần

Giá trị đồng hồ lô gic	Máy chủ 1	Máy chủ 2	Máy chủ 3
1	$S_1 \rightarrow S_2$:REQ-CS,1,1 $S_1 \rightarrow S_3$:REQ-CS,1,1		
2		$S_2 \rightarrow S_1$:REQ-CS,2,2 $S_2 \rightarrow S_3$:REQ-CS,2,2	
3	$S_2 \rightarrow S_1$:REQ-CS,2,2	$S_1 \rightarrow S_2$:REQ-CS,1,1	$S_2 \rightarrow S_3$:REQ-CS,2,2
4			$S_3 \rightarrow S_2$:REP-CS,4,3
5		$S_3 \rightarrow S_2$:REP-CS,4,3	$S_1 \rightarrow S_3$:REQ-CS,1,1
6			$S_3 \rightarrow S_1$:REP-CS,6,3
7	$S_2 \rightarrow S_1$:REP-CS,5,2		
8	**Máy chủ S_1 vào miền găng**		
9	$S_1 \rightarrow S_2$:REP-CS,9,1		
10	$S_1 \rightarrow S_2$:REL-CS,10,1 $S_1 \rightarrow S_3$:REL-CS,10,1	$S_1 \rightarrow S_2$:REP-CS,9,1	
11	**Máy chủ S_2 vào miền găng**		

1. Cho phép chỉ có một tiến trình duy nhất được thực thi trong miền găng tại một thời điểm đối với một tài nguyên.
2. Nếu không có tiến trình nào trong miền găng, tiến trình yêu cầu vào miền găng phải được phép vào và thực thi trong khoảng thời gian cho phép.
3. Khi có nhiều tiến trình yêu cầu vào miền găng, việc cho phép có thể bị trì hoãn cho đến khi được cấp phép.
4. Tiến trình xử lý trong miền găng trong không bị chặn bởi các tiến trình khác.

Hai nhóm giải pháp chính trong loại trừ tương hỗ đó là:

Nhóm thứ nhất là phương pháp tiếp cận dựa trên token: Ricart-Agrawala [11], [120], Mizuno-Neilsen-Rao [91], Helary-Plouzeau-Raynal [45], Raymond [103], Singhal [116], Naimi-Trehel [92], Mishra-Srimani [90], Nishio [97].

Nhóm thứ hai là phương pháp tiếp cận dựa trên quyền: Lamport [71], Ricart-Agrawala [107], Carvalho-Roucairol [11], Raynal [89], Maekawa [83], Sanders [108], Agrawal-El Abbadi [1], Singhal [1].

Bảng 1.5: Hiệu năng của thuật toán dựa trên quyền

Thuật toán	**Tổng số thông điệp**
Lamport	$3(N$ - $1)$
Ricart-Agrawala	$2(N$ - $1)$
Carvalho-Roucairol	0 đến $2(N$ - $1)$
Raynal	$2(N-1)^2$
Maekawa	$3\sqrt{N}$ đến $5\sqrt{N}$
Sanders	$\|Ii-\{i\}\|+2\,(\|Ri-\{i\}\|)$
Agrawal-El Abbadi	O(log N)
Singhal	$(N$ - $1)$ đến $3(N$ - $1)/2$

Theo Kshemkalyani và Singhal trong [62] trình bày hiệu năng loại trừ tương hỗ được xác định dựa trên các tham số: độ phức tạp thông điệp, độ trễ quá trình đồng bộ hóa, thời gian hồi đáp và thông lượng hệ thống. Bên cạnh đó, hiệu năng loại trừ tương hỗ dựa trên điều kiện của tải trong hệ thống. Trong đó tải được xác định bởi tỷ lệ thông điệp đến yêu cầu thực thi miền găng. Đối với tải thấp, số lượng tiến trình phải đợi chờ để vào miền găng là rất thấp. Đối với tải cao, luôn có tiến trình yêu cầu thựcthi miền găng phải chờ trong hàng đợi. Khảo sát của Velazquez trong [129] đưa ra đánh giá hiệu năng của thuật toán dựa trên quyền thể hiện ở Bảng 1.5.

Bảng 1.6: Phân tích, so sánh hiệu năng của các thuật toán loại trừ tương hỗ

Thuật toán	**Thời gian hồi đáp**	**Độ trễ đồng bộ**	**Thông điệp tải thấp**	**Thông điệp tải cao**
Lamport	$2\,T + E$	T	$3(N$ - $1)$	$3(N$ - $1)$
Ricart-Agrawala	$2\,T + E$	T	$2(N$ - $1)$	$2(N$ - $1)$
Suzuki-Kasami	$2\,T + E$	T	N	N
Raymond	$T(\log N) + E$	$T(\log N)/2$	$\log(N)$	4

Yadav và cộng sự trong [138] trình bày tổng quan nghiên cứu loại trừ tương hỗ trong môi trường phân tán được chia thành hai nhóm chính là thuật toán tập trung

và thuật toán phân tán. Phân loại trong thuật toán loại trừ tương hỗ phân tán, hai giải pháp tiếp cận là giải pháp dựa trên nội dung và giải pháp dựa trên điều khiển. Đối với giải pháp dựa trên nội dung sử dụng thuật toán trật tự nhãn thời gian lô gic và thuật toán bầu chọn. Đối với giải pháp dựa trên điều khiển sử dụng cấu trúc cây, cấu trúc truyền broadcast và cấu trúc mạng vòng [96]. Yadav và cộng sự phân tích, so sánh hiệu năng của các thuật toán loại trừ tương hỗ phân tán thể hiện trong Bảng 1.6.

Dựa trên khảo sát, đánh giá hiệu năng Bảng 1.5 và Bảng 1.6 trong [129, 138], độ phức tạp thông điệp hai thuật toán loại trừ tương hỗ phân tán dựa trên quyền Lamport [71] là $3(N - 1)$ và Ricart-Agrawala [107] là $2(N - 1)$ trong trường hợp tải cao. Hướng nghiên cứu của Luận án trình bày giải pháp song song hóa thuật toán Lamport trong loại trừ tương hỗ phân tán để giảm độ phức tạp thông điệp, thời gian hồi đáp và độ trễ đồng bộ thể hiện:

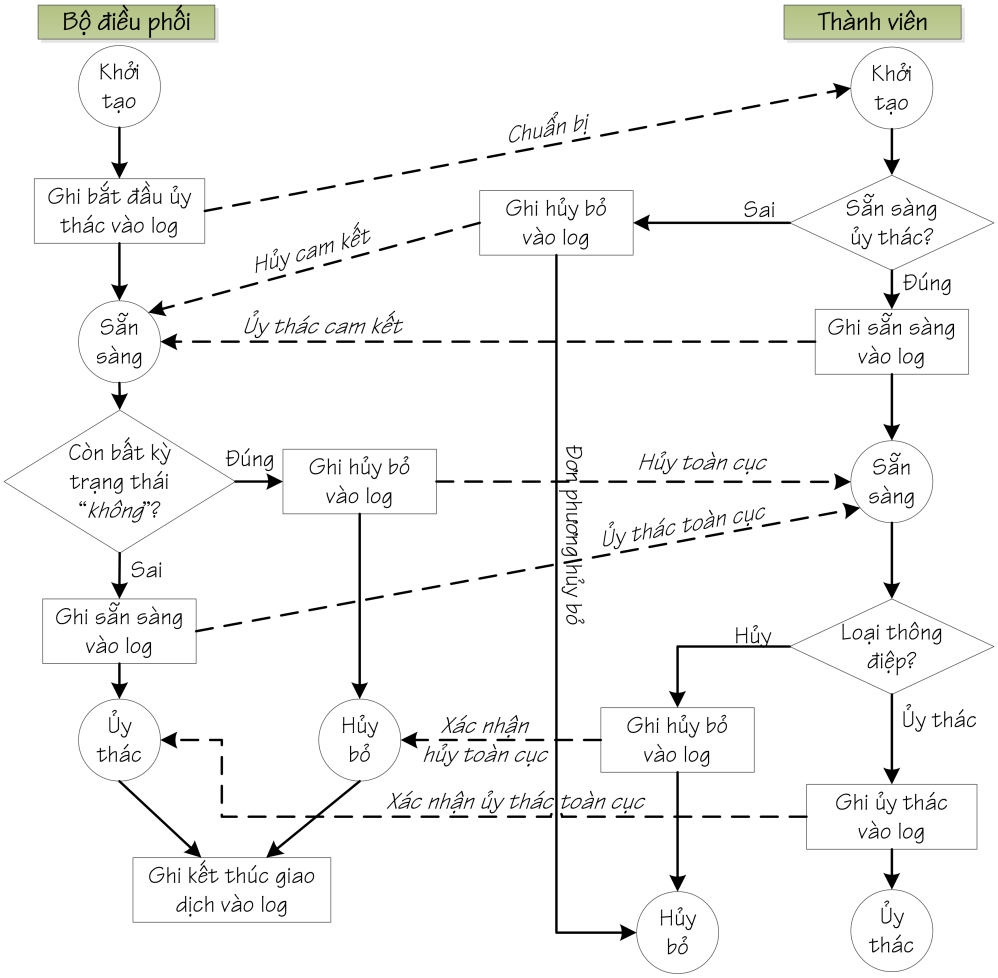
1. *Đồng bộ hóa các tiến trình dựa trên song song hóa thuật toán Lamport đạt được trật tổng quát chặt chẽ với độ phức tạp thông điệp 3(N - 1). Giải pháp được trình bày trong Mục 2.1.1.*

2. *Các máy chủ hợp lực để vào miền găng thời điểm sớm hơn trong thuật toán loại trừ tương hỗ phân tán với thời gian hồi đáp là (N - 1) + E, độ phức tạp thông điệp là (N - 1) và độ trễ đồng bộ T = 0. Giải pháp được trình bày trong Mục 2.1.2.*

1.2.1.4. Vấn đề kỹ thuật cung cấp tài nguyên dùng chung trong hệ phân tán

Theo Chương 14 trong [62] của tác giả Kshemkalyani và Singhal và theo Chương IV, V [65] của tác giả Lê đề cập đến nguyên lý hoạt động hệ phân tán dựa vào *sự hợp lực giữa các máy chủ* trong hệ để cung cấp tài nguyên dùng chung. Thông qua sự hợp lực, tài nguyên dùng chung được cung cấp duy nhất đối với người sử dụng. Để đạt được trạng thái này, các máy chủ xử lý tài nguyên dưới sự điều khiển thống nhất của hệ điều hành gọi là *hệ điều hành phân tán*.

Một trong những nhược điểm của hệ phân tán là phụ thuộc vào môi trường truyền thông có độ trễ. Khi tài nguyên dùng chung gần tới hạn thì quá trình tranh chấp tài nguyên diễn ra càng cao vì số lượng tiến trình truy cập vào tài nguyên lớn hơn nhiều so với khả năng cung cấp, trạng thái này gọi là *tương tranh tài nguyên*. Bên cạnh đó, một tiến trình đang chiếm giữ tài nguyên lại phát đi yêu cầu tài nguyên mà đang được chiếm giữ bởi tiến trình khác, hiện tượng này tạo thành vòng khép kín gọi là *bế tắc* trong cung cấp tài nguyên.



Hình 1.8: Hành động giao thức 2 pha 2PC [99]

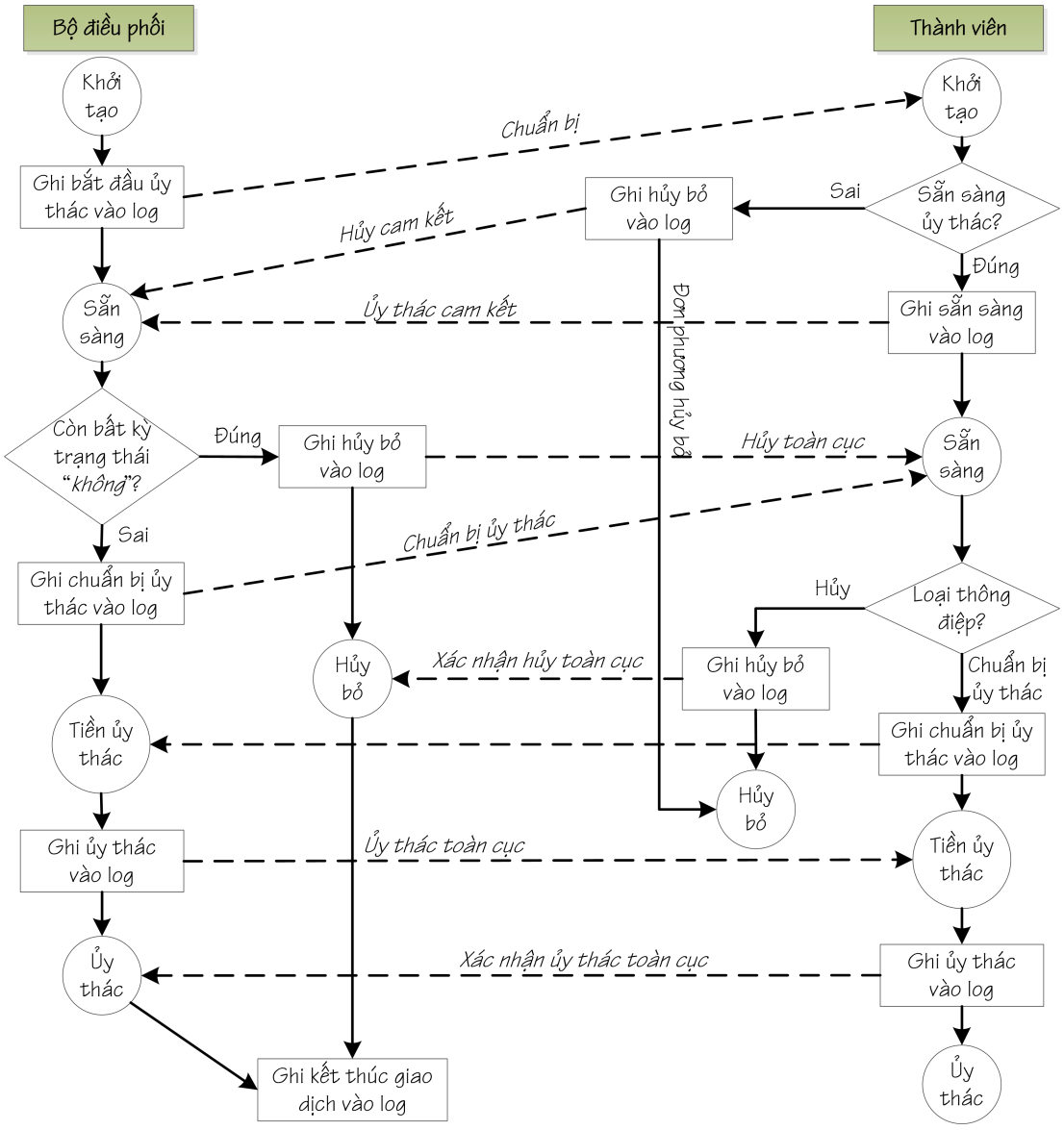
Do đó, để cung cấp tài nguyên dùng chung nhất thiết phải xây dựng giải pháp nhằm phòng, tránh tương tranh và bế tắt. Giải pháp cung cấp tài nguyên trong hệ phân tán đề xuất được xây dựng dựa trên các đặc tính cơ bản của hệ phân tán; trong đó, *tính gắn bó* được xem là một trong những đặc tính quan trọng của hệ. Thông qua tính gắn bó, tài nguyên được cung cấp duy nhất trên tất cả các máy chủ trong hệ. Tùy thuộc mức độ và yêu cầu ứng dụng để triển khai tính gắn bó trong hệ phân tán.

Theo *Chương 12: Nguyên lý các hệ cơ sở dữ liệu phân tán* của tác giả Özsu và Valduriez trong [99] đề cập giải pháp nhất quán dữ liệu trên các máy chủ thông qua các giao dịch phân tán dựa trên giao thức 2 pha 2PC (two-phase commit protocol) theo Hình 1.8 và 3 pha 3PC (three-phase commit protocol) theo Hình 1.9.

Trong 2PC theo Hình 1.8 có hai nhược điểm chính:

1. Giao dịch mất nhiều thời gian chờ kết thúc, nút điều phối và nút tham gia chờ thông điệp gửi đến để hoàn tất hay hủy bỏ giao dịch.

2. Bộ điều phối bị lỗi sẽ dẫn đến ảnh hưởng toàn bộ hệ thống.



Hình 1.9: Hành động giao thức 3 pha 3PC [99]

Giải pháp khác là giao thức 2 pha tuyến tính hay còn gọi là giao thức lồng 2 pha trình bày bởi Gray trong [40] và giao thức 2 pha phân tán. Nhược điểm của các giao thức là các giao dịch trước đó không nằm trong trạng thái sẵn sàng ủy thác và trật tự tổng quát chặt chẽ không được thiết lập. Điều này làm tăng thời gian giao dịch để cập nhật dữ liệu và triển khai giao dịch phân tán khó hơn.

3PC ở Hình 1.9 khác với 2PC ở Hình 1.8 là có thêm pha tiền ủy thác trước khi ủy thác. Pha tiền ủy thác cho phép chờ đợi, kiểm soát các giao dịch đang thực hiện trên các máy chủ trong hệ cho đến khi hoàn tất pha này. Nếu một trong số máy chủ

thực hiện không thành công, lập tức hủy bỏ toàn bộ giao dịch. Nhược điểm của 3PC là giao dịch được ủy thác nhưng không khóa (non-blocking), bên cạnh đó quá trình trao đổi thông điệp quá nhiều để khôi phục độc lập dữ liệu dẫn đến chi phí tài nguyên truyền thông cao. Theo tác giả Manoj Kumar trong [63] trình bày so sánh giữa 2PC và 3PC thể hiện qua Bảng 1.7:

Về mặt thuận lợi, 2PC đảm bảo gắn bó chặt chẽ hơn, trao đổi thông điệp ít hơn, chi phí truyền thông thấp và độ phức tạp thấp hơn so với 3PC. Nhưng ngược lại, 2PC sử dụng khóa trong giao dịch thấp, hiệu năng cho các giao dịch phân tán thấp và khó khăn trong quá trình triển khai, vận hành.

Theo *Chương IV: Nguyên lý thiết kế các chiến lược đồng bộ hóa các tiến trình* trong [65] của tác giả Lê đã đưa ra các chiến lược cung cấp tài nguyên trong hệ phân tán. Chiến lược này nhằm đảm bảo tính gắn bó nhiều bản sao trên cơ sở nhờ dấu như các thuật toán: Lamport, Lomet, Le Lann, Ellis, Mullery,... Dựa vào các nguyên lý của hệ phân tán, các nghiên cứu mở rộng về cơ sở dữ liệu phân tán kết hợp giao dịch có khóa và dấu để đảm bảo tính nhất quán trong cơ sở dữ liệu đồng thời điều khiển tương tranh bằng dấu.

Bảng 1.7: So sánh 2PC và 3PC

	2 pha	**3 pha**
Loại giao dịch	2 pha	2 pha mở rộng
Pha bầu chọn	x	x
Pha tiền ủy thác	-	x
Pha ủy thác	x	x
Khóa giao dịch	thấp	cao
Trao đổi thông điệp	$4(n\text{-}1)$	$5(n\text{-}1)$
Chi phí truyền thông	thấp	cao
Ghi log	$2n$	$2n$
Độ phức tạp	$O(n^2)$	$O(n^3)$
Hiệu năng	thấp	cao
Áp dụng cho giao dịch phân tán	khó khăn	dễ dàng

Trên cơ sở đó, nghiên cứu của Luận án tập trung vào *giải pháp đảm bảo gắn bó mạnh* cho các ứng dụng đăng ký tài nguyên trực tuyến dựa vào cơ chế quản lý nhiều bản sao. Đối với hệ thống đăng ký tài nguyên trực tuyến, mỗi máy chủ có cơ sở dữ liệu riêng DB_i cho máy chủ của mình. Mục đích chính cho giải pháp đảm bảo gắn bó mạnh là mỗi hành động diễn ra hay cập nhật tại máy chủ nào đồng thời

cập nhật tên tất cả các máy chủ còn lại trong hệ thống để đảm bảo tính nhất quán $DB_1 = DB_2 = \ldots = DB_n$. *Vì vậy, giải pháp đảm bảo gắn bó là một trong những giải pháp cần thiết để nghiên cứu và xây dựng trong hệ phân tán thông qua cơ chế truyền thông điệp, giải pháp đảm bảo gắn bó mạnh được trình bày trong Mục 2.2 của Luận án.*

1.2.2. Các nghiên cứu liên quan đến điều khiển cung cấp tài nguyên trong hệ thống ảo hóa

Khảo sát của các tác giả Vinothina, Sridaran và Ganapathi trong [131] trình bày tổng quan về chiến lược cung cấp tài nguyên trong Điện toán đám mây. Thảo luận về chiến lược này đề cập đến tích hợp hoạt động cung cấp đối với các tiện ích và dịch vụ trong điều kiện tài nguyên giới hạn để đáp ứng nhu cầu người sử dụng và các ứng dụng đám mây. Các tác giả cũng trình bày tóm tắt phân loại cung cấp tài nguyên và tác động của nó trong hệ thống đám mây. Các khuyến cáo đưa ra trong nghiên cứu đối với chiến lược cung cấp tài nguyên đó là:

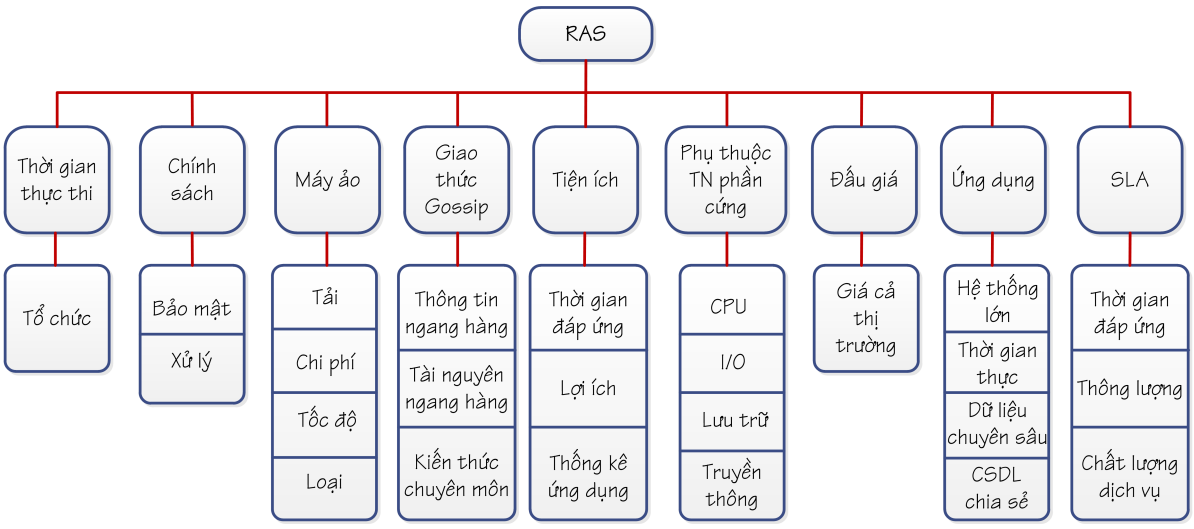
- Trường hợp tương tranh tài nguyên phát sinh khi hai hay nhiều ứng dụng cố gắng truy cập vào nguồn tài nguyên dùng chung tại cùng một thời điểm.
- Thiếu tài nguyên phát sinh khi tài nguyên là hữu hạn.
- Tình trạng phân mảnh tài nguyên phát sinh khi các tài nguyên đang bị cô lập.
- Vượt quá dự phòng tài nguyên phát sinh khi ứng dụng nhận tài nguyên dư thừa hơn so với yêu cầu.
- Không đủ dự phòng tài nguyên xảy ra khi ứng dụng được gán ít tài nguyên hơn so với nhu cầu.

Ngoài ra, trong khảo sát của các tác giả Vinothina, Sridaran và Ganapathi cũng trình bày sơ đồ mô tả phân loại các chiến lược cung cấp tài nguyên đề xuất trong mô hình Điện toán đám mây theo Hình 1.10. Trên cơ sở thảo luận từng loại trong sơ đồ, các tác giả đã trình bày các thuận lợi và hạn chế của việc cung cấp tài nguyên trong đám mây. Các thuận lợi của việc cung cấp tài nguyên đám mây:

- Người sử dụng không phải cài đặt phần mềm hoặc phần cứng để truy cập các ứng dụng, phát triển và lưu trữ ứng dụng thông qua Internet.
- Không có giới hạn về địa điểm và phương tiện, người sử dụng có thể tiếp cận các ứng dụng và dữ liệu ở mọi nơi trên thế giới, trên bất kỳ phương tiện có kết nối đến hệ thống.

- Người sử dụng không cần phải tốn nhiều tiền cho việc mở rộng dịch vụ, phần cứng và phần mềm.
- Các nhà cung cấp dịch vụ đám mây có thể chia sẻ tài nguyên của họ qua Internet trong điều kiện khan hiếm tài nguyên.

Nghiên cứu của Kong trong [58], Urgaonkar và các cộng sự trong [128], Sharkh và các cộng sự trong [112] trình bày các nghiên cứu về giải pháp ảo hóa tập trung vào trung tâm dữ liệu (Data Center). Nghiên cứu về tính toán lưới (Grid Computing) đề cập bởi hai nhóm tác giả Xhafa, Abraham trong [93] và Murphy, Goasguen trong [136]. Nhóm tác giả Velte và cộng sự trong [75], Lee và Zomaya trong [130], Pandey và các cộng sự trong [100], Sontakke và các cộng sự trong [117], Yan, Jinkuan và Jinghao trong [139] trình bày những chiến lược mới trong cung cấp tài nguyên. Các giải pháp nghiên cứu thích ứng với kế thừa được sử dụng trong các môi trường Điện toán đám mây. Ngoài ra, các giải pháp tối ưu trong cung cấp tài nguyên cho hệ thống đám mây đề cập bởi các tác giả Farooq và Zhu trong [33], Subhash và Thooyamani trong [119], Akintoye và Bagula trong [3].



Hình 1.10: Các chiến lược cung cấp tài nguyên trong Điện toán đám mây [131]

Cách tiếp cận để xử lý cung cấp tài nguyên truyền thông trên Điện toán đám mây dựa trên quan điểm là ảo hóa mạng. Cung cấp tài nguyên dựa trên môi trường mạng ảo đang được các nhà khoa học quan tâm nghiên cứu. Nhóm nghiên cứu Buyya và cộng sự trong [8] trình bày các giải pháp tập trung tối ưu hóa tiêu thụ băng thông. Môi trường truyền thông được ảo hóa bởi mạng ảo được trình bày chi tiết bởi các tác giả Mosharaf và Boutaba trong nghiên cứu [18] để đảm bảo các kết nối các máy chủ ảo trong đám mây.

Trong quá trình cung cấp tài nguyên, hệ thống cung cấp tài nguyên có thể đạt được mức tiêu thụ băng thông tối ưu bằng cách quan sát tiêu thụ băng thông trong thời gian thực và đưa ra quyết định về tái cung cấp. Đơn cử cho trường hợp này là các máy ảo di trú để giải phóng được các máy chủ sử dụng hay các dữ liệu lớn di chuyển qua các máy ảo. Một mạng ảo có thể kết hợp với một ứng dụng phát triển bao gồm một số máy chủ và cơ sở dữ liệu cũng như các mạng kết nối giữa chúng.

Quản lý tiêu thụ băng thông là tiêu chí quan trọng thiết kế toàn cục trong Điện toán đám mây. Theo nghiên cứu trong [8], Buyya và cộng sự trình bày việc tối ưu băng thông là một vấn đề thách thức và phức tạp. Nguyên nhân chính là các ứng dụng tính toán và dữ liệu được phát triển một cách nhanh chóng trong khi các máy chủ yêu cầu tính toán ngày càng lớn và ổ đĩa cần thiết để xử lý chúng đủ nhanh trong khoảng thời gian nhỏ nhất. Tuy nhiên, tiết kiệm băng thông có thể đạt được thông qua nhiều chiến lược, như tận dụng tối đa tài nguyên hiện tại, tắt hoặc thiết lập CPU ngủ đông khi máy chủ trở nên nhàn rỗi, chuyển tải công việc,...

Nghiên cứu của Urgaonkar và các cộng sự trong [128] sử dụng hàng đợi thông tin để thực hiện cấp phát nguồn và cơ chế quản lý băng thông trong trung tâm dữ liệu ảo hóa. Trường hợp giả định rằng mỗi ứng dụng tham gia có một số lượng nhất định máy ảo được tổ chức cùng một số máy chủ vật lý. Hàng đợi thông tin các máy chủ được sử dụng để đưa ra quyết định điều khiển trực tuyến theo những thay đổi tải công việc của ứng dụng. Tiết kiệm băng thông được thực hiện bằng cách chuyển các máy chủ nhàn rỗi vào chế độ không hoạt động khi tải công việc của chúng là thấp, và kích hoạt lại khi tải công việc tăng lên.

Bộ cung cấp tài nguyên băng thông cho trung tâm dữ liệu ảo hóa được nghiên cứu đề xuất bởi Buyya trong [8], Nan và các cộng sự trong [95]. Các nghiên cứu này xem xét một môi trường mà các nhà phát triển đưa ra yêu cầu máy ảo trong đám mây. Trong trường hợp trên, các nhà cung cấp phải đối phó với sự cân bằng giữa băng thông và hiệu năng. Cụ thể, nhà phát triển phải được tiêu thụ băng thông tối thiểu trong khi đạt yêu cầu phát triển, việc giảm tiêu thụ băng thông được thực hiện bằng cách tắt máy chủ nhàn rỗi. Do đó, kiến trúc hệ thống ảo hóa được đề xuất bao gồm các bộ cung cấp, quản lý toàn cục và quản lý cục bộ.

Tóm lại, cung cấp tài nguyên trong môi trường đám mây cần có những giải pháp, chiến lược nhằm tối ưu tài nguyên cho người sử dụng và các ứng dụng đám mây. Trong chiến lược cung cấp tài nguyên truyền thông cho đám mây của Luận án, việc xây dựng cơ chế và điều khiển cung cấp tài nguyên là đặc điểm quan trọng để đảm bảo tính gắn bó cho hệ phân tán, di trú giữa các đám mây và thời gian thực đối với các ứng dụng.

1.2.2.1. Ảo hóa mạng

Ảo hóa mạng là tiến trình kết hợp các nguồn tài nguyên phần cứng mạng và tài nguyên phần mềm mạng thành một đơn vị quản lý duy nhất. Mục tiêu của ảo hóa mạng là cung cấp cho hệ thống một cách hiệu quả với người sử dụng, điều khiển và bảo mật chia sẻ các tài nguyên mạng được trình bày bởi tác giả Rees trong nghiên cứu [106].

Theo báo cáo của VMware [125] trình bày mô hình ảo hóa mạng thể hiện trong Hình 1.11, sản phẩm cuối cùng của ảo hóa mạng là *mạng ảo*. *Mạng ảo* được phân thành hai loại: bên ngoài và nội bộ. Mạng ảo bên ngoài bao gồm một số mạng cục bộ LAN (Local Area Network) được quản lý bởi phần mềm như một thực thể duy nhất. Xây dựng các khối mạng ảo bên ngoài cơ bản là chuyển đổi phần cứng và kỹ thuật phần mềm mạng ảo cục bộ VLAN (Virtual Local Area Network). Ví dụ về các mạng ảo ngoài bao gồm các mạng cộng tác lớn và các trung tâm dữ liệu. Để kiểm soát, thực thi và quản lý ảo hóa mạng, một phương pháp để tiếp cận là SDN (Software Defined Networking) được nghiên cứu và trình bày trong [48, 51, 77]. SDN cho phép xây dựng và phát triển chương trình tham gia vào quá trình điều khiển và tối ưu truyền thông, đặc biệt đối với việc xây dựng giải pháp mã mạng cho hệ thống máy ảo.

Mạng ảo nội bộ bao gồm một hệ thống sử dụng máy ảo hoặc vùng có giao diện mạng được cấu hình trên ít nhất một card giao tiếp mạng vật lý NIC (Network Interface Card). Những giao diện mạng được gọi là card giao diện mạng ảo hoặc NIC ảo VNIC (Virtual Network Interface Card). Những giao diện này có thể giao tiếp với nhau như đang cùng trên mạng cùng một cục bộ, hiệu quả là trở thành một mạng ảo trên một máy chủ duy nhất.

Một loại đặc biệt của mạng ảo nội bộ là mạng riêng ảo VPN (Virtual Private Network). Các mạng riêng ảo này khác với các mạng riêng ảo khác. Phần mềm VPN tạo ra một liên kết an toàn point-to-point giữa hai hệ thống thiết bị đầu cuối. VPN là một mạng ảo trên một hệ thống mà không được truy cập bởi hệ thống bên ngoài. Sự độc lập của mạng nội bộ này từ hệ thống bên ngoài khác có thể đạt được bằng cách cấu hình VNIC qua etherstubs.

Các nhà triển khai có thể kết hợp các nguồn tài nguyên mạng để cấu hình cả hai mạng ảo nội bộ và bên ngoài. Ví dụ, các nhà triển khai có thể cấu hình hệ thống cá nhân với mạng ảo nội bộ vào mạng LAN là một phần của hệ thống lớn, mạng ảo bên ngoài.

Hình 1.11: Mô hình ảo hóa mạng [125]

org.cloudbus.cloudsim.network.datacenter là gói thuộc Cloudsim áp dụng cho mạng ảo mà Luận án kế thừa và phát triển. Gói *network* cho phép các nhà nghiên cứu thêm các thành phần và các thuật toán mở rộng và tối ưu mạng ảo trong hệ thống ảo hóa. Các thành phần bên trong gói được mô tả trong Bảng 1.8. Đối với Luận án, các lớp phát triển theo các thuật toán của hệ phân tán theo thứ tự 2, 4, 6, 8, 9, 10 trong Bảng 1.8. Đối với giải pháp kỹ thuật mã mạng, ngoài các lớp sử dụng trong hệ phân tán, các lớp còn lại được phát triển và mở rộng thêm trong Bảng 1.8 cho thuật toán điều khiển cung cấp tài nguyên truyền thông.

1.2.2.2. Mạng điều khiển bằng phần mềm (SDN) và ảo hóa chức năng mạng (NFV-Network Functions Virtualization)

SDN là một cấu trúc trong hệ thống ảo hóa được thiết kế cho phép hệ thống mạng trở nên linh động và có hiệu quả chi phí truyền thông. Đây là một kiến trúc linh hoạt, có thể quản lý, hiệu quả và thích nghi, lý tưởng cho kết nối hệ thống băng thông rộng, đáp ứng yêu cầu về tài nguyên truyền thông cho các ứng dụng đám mây. Kiến trúc SDN tách các chức năng điều khiển và chuyển tiếp mạng, cho phép lập trình trực tiếp cho các ứng dụng và dịch vụ mạng. Việc xây dựng các giải pháp SDN đang tập trung phát triển bộ giao thức OpenFlow.

Bảng 1.8: Gói *network* trong trung tâm dữ liệu của Cloudsim

TT	Lớp	Mô tả
1	AggregateSwitch	Lớp AggregateSwitch cho phép mô phỏng thiết bị chuyển mạch tổng cho mạng trung tâm dữ liệu.
2	AppCloudlet	Lớp AppCloudlet trình bày ứng dụng cho người dùng để thực thi trong trung tâm dữ liệu.
3	EdgeSwitch	Lớp này cho phép mô phỏng thiết bị chuyển mạch có cấu hình cố định cho mạng Datacenter.
4	HostPacket	Lớp HostPacket trình bày gói tin đi qua mạng ảo với một Host.
5	NetDatacenterBroker	Lớp NetDatacentreBroker trình bày hoạt động bộ môi giới đại diện cho bộ cung cấp tài nguyên trung tâm dữ liệu.
6	NetworkCloudlet	Lớp NetworkCloudlet là lớp mở rộng của lớp Cloudlet để hỗ trợ mô phỏng các ứng dụng phức tạp.
7	NetworkCloudletSpaceSharedScheduler	Lớp CloudletSchedulerSpaceShared thực thi chính sách lập lịch được thực hiện bởi một máy ảo.
8	NetworkDatacenter	Lớp NetworkDatacenter là một trung tâm dữ liệu có danh sách các máy chủ (hostList) được ảo hóa và nối mạng.
9	NetworkHost	Lớp NetworkHost là lớp mở rộng của lớp Host để hỗ trợ mô phỏng các trung tâm dữ liệu đã được nối mạng.
10	NetworkPacket	Lớp NewtorkPacket trình bày gói tin di chuyển từ máy chủ này sang máy chủ khác.
11	NetworkVm	Lớp NetworkVm là lớp mở rộng của lớp Vm để hỗ trợ mô phỏng các trung tâm dữ liệu đã được nối mạng.
12	NetworkVmAllocationPolicy	Lớp NetworkVmAllocationPolicy là chính sách cung cấp tài nguyên mạng cho máy ảo.
13	RootSwitch	Lớp RootSwitch cho phép mô phỏng thiết bị chuyển mạch gốc kết nối trung tâm dữ liệu với mạng bên ngoài.
14	TaskStage	Lớp Taskstage trình bày các giai đoạn tác vụ khác nhau mà networkCloudlet có thể có trong quá trình thực thi.
15	WorkflowApp	Lớp WorkflowApp là một ví dụ về AppCloudlet có ba tác vụ giao tiếp.

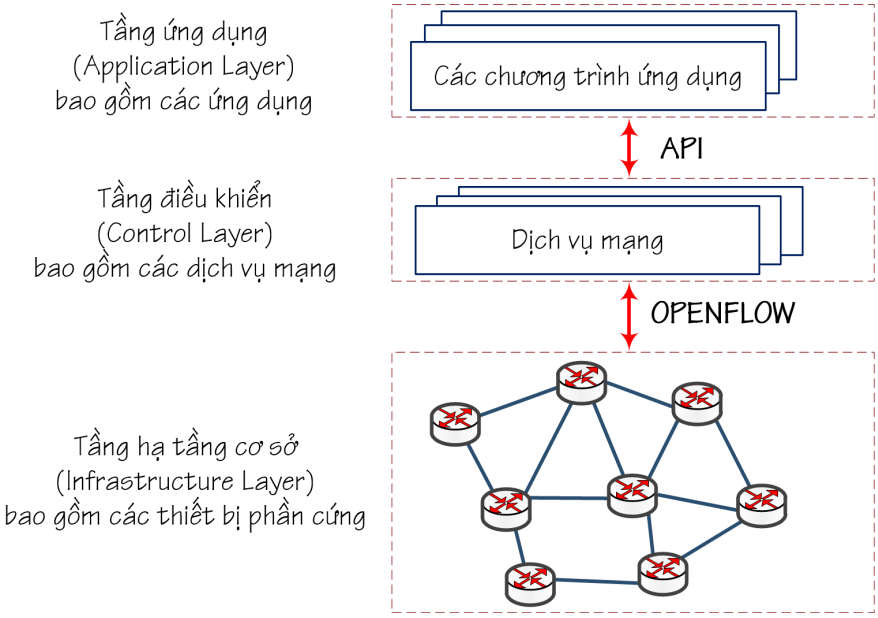
Theo Hu, Hao và Bao [49], cấu trúc của SDN bao gồm ba tầng theo mô tả ở Hình 1.12. Giao thức OpenFlow tiếp nhận luồng dữ liệu từ tầng điều khiển để chuyển và định tuyến các gói tin trong hệ thống mạng thông qua tầng hạ tầng cơ sở, giao thức này cho phép lập lịch và điều khiển lưu lượng mạng. API (Application Programming Interface) cho phép chuyển dữ liệu các ứng dụng từ tầng ứng dụng xuống dịch vụ mạng ở tầng điều khiển, các API cho phép thiết lập, giám sát lưu lượng và triển khai dịch vụ.

Dựa vào cấu trúc trên, SDN đang được quan tâm nghiên cứu và công bố của các hãng lớn như VMWare, Cisco trong [87], HP trong [48],...

Trong cơ chế ảo hóa, bên cạnh SDN có thành phần đi kèm đó là NFV. Các nghiên cứu về NFV của hai nhóm tác giả Cotrone và cộng sự [20], Han và cộng sự trong [43]

trình bày việc ảo hóa các chức năng mạng như: tường lửa, cơ chế NAT, cân bằng tải,... để đạt được tính linh động và thúc đẩy việc triển khai cung cấp dịch vụ mạng.

Hai thành phần SDN và NVF kết hợp với nhau tạo thành bộ đôi để phát triển kỹ thuật ảo hóa mạng cho hệ thống Điện toán đám mây. Đây là cơ hội đồng thời là thách thức đối với việc chuyển đổi từ hạ tầng vật lý sang hạ tầng ảo hóa, các thách thức trình bày trong [8, 18, 39, 42, 111] cho hệ thống mạng ảo tập trung vào các vấn đề: tắc nghẽn, mở rộng thành phần, điều khiển, hiệu năng và bảo mật.



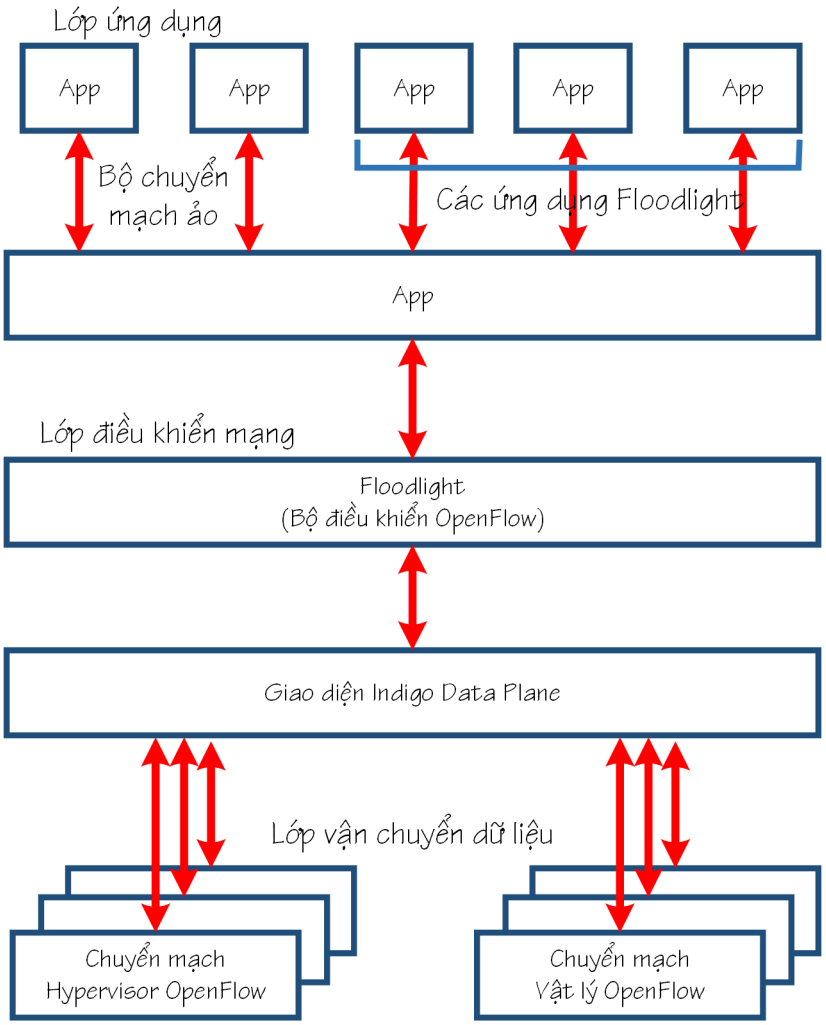
Hình 1.12: Cấu trúc của SDN [49]

Qua các khái niệm và định nghĩa về Điện toán đám mây, rõ ràng rằng các máy chủ ảo mang lại hiệu quả và tối ưu hiệu năng cho hệ thống vật lý sẵn có. Điều này dẫn đến nhiều máy ảo sẽ tranh chấp tài nguyên vật lý, nhất là đối với tài nguyên truyền thông. Tài nguyên truyền thông vốn hữu hạn so với khả năng cung cấp, do đó dẫn đến bế tắc khi nhu cầu lưu lượng tăng cao bởi các máy chủ ảo. Các nghiên cứu trình bày trong [13, 22, 23, 144] nêu lên các giải pháp nhằm xây dựng cơ chế điều khiển giảm bế tắc trong ảo hóa mạng.

Trên các hệ thống ảo hóa, ví dụ như hệ thống VMWare, Virtualbox và Qemu có cài đặt sẵn mã nguồn mở bộ điều khiển SDN/Openflow. Bộ điều khiển SDN/Openflow trên các hệ thống ảo hóa là OpenDayLight, ONOS, Ryu và Mininet. Trong đó Mininet rất cần thiết, vì nó tạo ra các tô pô chuyển mạch giao thức Openflow, Mininet như một máy ảo độc lập và tương tác thông qua giao diện MiniEdit. Theo Floodlight[1],

[1]http://www.projectfloodlight.org/floodlight

Floodlight SDN Controller là công cụ hỗ trợ lập trình mở rộng cho Mininet, cấu trúc của Floodlight thể hiện qua Hình 1.13.



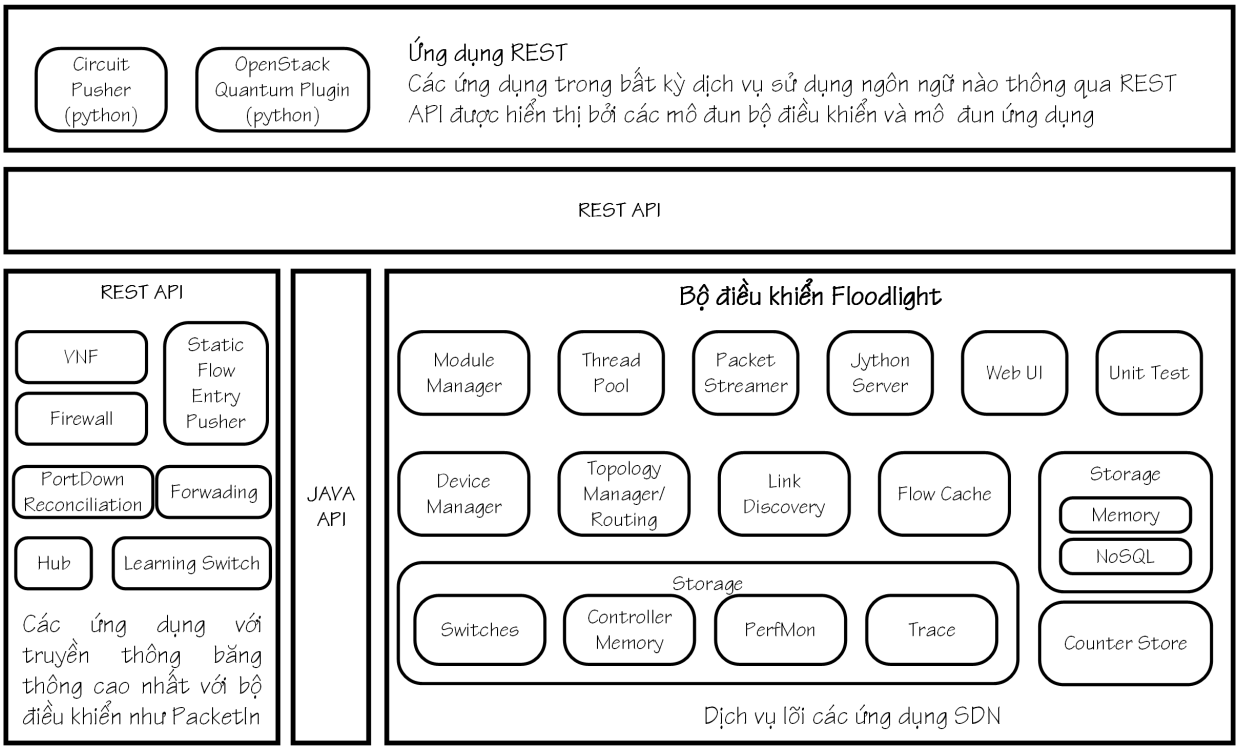
Hình 1.13: Cấu trúc của Floodlight

Tổ chức Open Networking Foundation (ONF) để thúc đẩy công nghệ OpenFlow và giao thức chuyển mạch OpenFlow Switching Protocol. Theo Hình 1.13, Floodlight là bộ điều khiển mở giao thức Openflow, hoạt động với bộ chuyển mạch vật lý và ảo. Floodlight được cấp phép bởi Apache, cho phép sử dụng Floodlight với hàu hết các mục đích mở rộng. Floodlight được hỗ trợ bởi một cộng đồng các nhà phát triển bao gồm một số kỹ sư từ Big Switch Networks. Floodlight được thiết kế để hoạt động với thiết bị chuyển mạch, bộ định tuyến, chuyển mạch ảo và điểm truy cập hỗ trợ chuẩn giao thức OpenFlow ngày càng tăng.

Tính năng nổi bật của Floodlight:

- Cung cấp một hệ thống mô đun tải giúp đơn giản để mở rộng và nâng cao
- Dễ dàng thiết lập
- Hỗ trợ các chuyển mạch ảo và vật lý trên giao thức OpenFlow
- Có thể điều khiển các mạng giao thức OpenFlow hỗn hợp và các mạng phi giao thức OpenFlow - nó có thể quản lý nhiều nhóm thiết bị phần cứng chuyển mạch
- Được thiết kế để có hiệu năng cao - hoạt động trên cơ chế đa luồng
- Hỗ trợ cho nền tảng đám mây điều phối OpenStack

Về mặt kiến trúc, theo Hình 1.13, Floodlight không những là bộ điều khiển giao thức OpenFlow ("Bộ điều khiển Floodlight") mà còn có tập hợp các ứng dụng được xây dựng phần trên Bộ điều khiển Floodlight.



Hình 1.14: Mối quan hệ giữa Bộ điều khiển Floodlight

Floodlight hiện thực hóa một tập hợp các chức năng phổ biến để kiểm soát và truy vấn mạng OpenFlow, trong khi các ứng dụng trên đó nhận ra các tính năng khác nhau để giải quyết các nhu cầu khác nhau của người dùng qua mạng. Hình 1.14 trình bày mối quan hệ giữa Bộ điều khiển Floodlight, các ứng dụng được xây dựng dưới

dạng các mô đun Java được biên dịch bằng Floodlight và các ứng dụng được xây dựng qua REST API của Floodlight.

Đối với hỗ trợ tô pô, Floodlight hiện có hai hoạt động ứng dụng chuyển tiếp gói, có các hành vi khác nhau và hoạt động với tô pô được mô tả như sau:

Forwarding: Được bật theo mặc định ngay lập tức khỏi hộp, Forwarding cho phép chuyển tiếp gói đầu cuối giữa bất kỳ hai thiết bị nào trong các tô pô mạng sau:

Trong một cụm OpenFlow: Khi bất kỳ thiết bị A nào gửi một gói tới thiết bị B trong cùng một cụm OpenFlow, Forwarding sẽ tính toán một con đường ngắn nhất giữa A và B. Các cụm OpenFlow có các cụm phi OpenFlow nằm giữa / trong số chúng: Mỗi cụm OpenFlow có thể có chính xác một liên kết kết nối với một cụm phi OpenFlow. Ngoài ra, các cụm OpenFlow và phi OpenFlow cùng nhau có thể không tạo thành các vòng lặp. Mỗi thiết bị sẽ có một điểm đính kèm trên mỗi cụm OpenFlow. Forwarding tính toán một con đường ngắn nhất trong mỗi cụm OpenFlow và chờ đợi các gói sẽ được chuyển tiếp ở các cụm phi OpenFlow. Các đường dẫn được cài đặt bằng cách hết thời gian chuyển tiếp khi không có lưu lượng nào được gửi qua đường dẫn lớn hơn khoảng thời gian chờ đã chỉ định (mặc định 5 giây).

Learning Switch: Learning Switch L2 đơn giản

- để sử dụng trong bất kỳ số lượng cụm OpenFlow nào, ngay cả với các cụm L2 phi OpenFlow ở giữa;
- không thể hoạt động nếu các bộ chuyển mạch trong tô pô lặp cụm;
- hiệu năng chuyển tiếp kém hơn nhiều so với các phương pháp khác.

Ngoài ra, Floodlight cũng cung cấp ứng dụng Static Flow Entry Pusher và ứng dụng Circuit Pusher cho phép người dùng chủ động cài đặt các đường dẫn chuyển tiếp trong mạng.

Static Flow Entry Pusher cho phép cài đặt chuyển luồng bằng bộ chuyển mạch, do đó tạo đường dẫn chuyển tiếp dựa trên lựa chọn về cổng chuyển mạch của người dùng. CircuitPizer xây dựng dựa trên Static Flow Entry Pusher, bộ quản lý thiết bị và dịch vụ định tuyến dựa trên REST API để xây dựng các mạch đường dẫn ngắn nhất trong một cụm OpenFlow.

Về mặt phát triển, Floodlight cho phép người lập trình có thể xây dựng các chương trình, thuật toán để thử nghiệm dựa trên ngôn ngữ Java. Các chương trình phát triển có thể tham gia vào tất cả các thành phần trong Bộ điều khiển Floodlight

ở Hình 1.14. Các mô đun có thể phát triển trong Bộ điều khiển Floodlight thể hiện:

- các địch vụ Floodlight
- Mô đun Bộ điều khiển: FloodlightProvider, DeviceManagerImpl, LinkDiscoveryManager, TopologyService, RestApiServer, ThreadPool, MemoryStorageSource, Packet Streamer, High Availability Support, Flow Cache (API only), OFSwitchManager
- Mô đun ứng dụng: VirtualNetworkFilter (Quantum Plugin), Forwarding, Firewall, Port Down Reconciliation, ACL (Access Control List)
- Mô đun tải hệ thống

Thử nghiệm Floodlight với Floodlight-Test. Floodlight-Test là một framework thực hiện thử nghiệm được phát hành với Floodlight để các nhà phát triển thực hiện các thử nghiệm tích hợp cho Floodlight và bất kỳ tiện ích mở rộng nào được thêm cho nhà phát triển. Floodlight-Test cho phép các nhà phát triển:

- Khởi tạo một hoặc nhiều VM (min) bằng mininet
- Chạy Floodlight trên máy chủ của nhà phát triển hoặc trong máy ảo
- Chạy một bộ kiểm tra tích hợp cơ bản được cung cấp
- Thêm thử nghiệm tích hợp mới cho mọi tiện ích mở rộng mới được phát triển

Floodlight-Test được phát hành để đảm bảo chất lượng cao của Floodlight và bất kỳ phần mở rộng nào của nó. Floodlight-Test khuyến khích và giúp các nhà phát triển tuân theo các phương pháp thử nghiệm phù hợp trong quy trình thiết kế của họ, đồng thời nó cũng đóng vai trò là tiêu chuẩn chất lượng cho các đóng góp của cộng đồng vào kho lưu trữ Floodlight. Theo thiết kế, Floodlight có nghĩa là một bộ điều khiển nguồn mở cho các ứng dụng OpenFlow và / hoặc các tính năng của bộ điều khiển được xây dựng.

Dựa trên những thuận lợi của Floodlight, định hướng triển khai bộ cung cấp tài nguyên dựa trên kỹ thuật mã mạng trên hệ thống ảo hóa trong thực tế là xây dựng chương trình dựa trên ngôn ngữ Java và tích hợp vào Floodlight. Bên cạnh đó, chương trình được tích hợp với các gói hỗ trợ vSphere Java API. Để triển khai chương trình trên VMWare, các gói hỗ trợ cho Java để kết nối đến vSphere là sim25, vijava5, dom4j, ... Thông qua các gói hỗ trợ, gói tin từ hệ phân tán sẽ chuyển đến bộ cung cấp tài

nguyên mã mạng. Bộ cung cấp tài nguyên mã mạng đảm nhiệm vai trò mã hóa gói tin và truyền đến các máy ảo trong hoặc ngoài hệ thống máy ảo theo Hình 1.13.

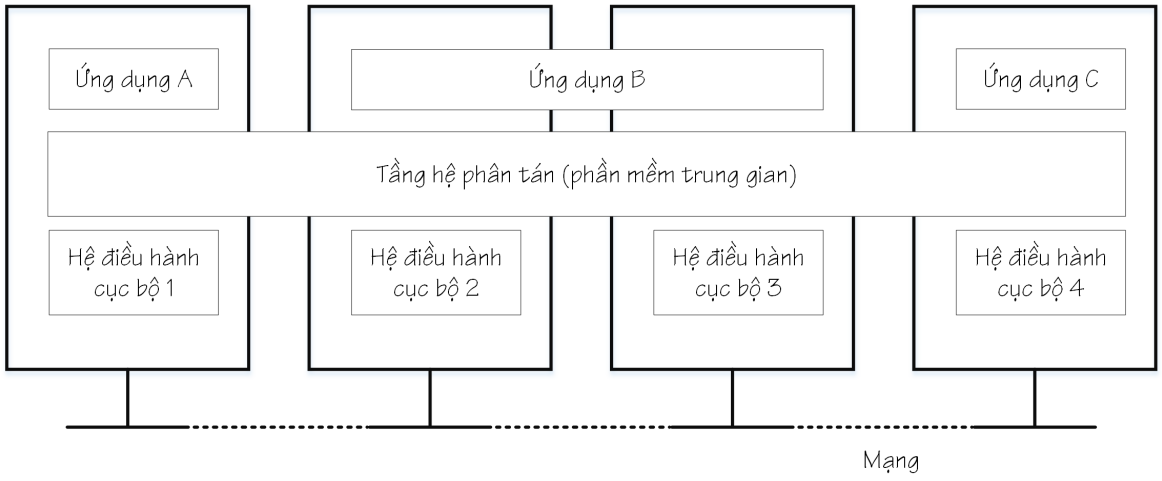
Nghiên cứu của Luận án tập trung vào giải pháp cung cấp tài nguyên truyền thông cho hệ thống ảo hóa; do đó, hướng nghiên cứu chính của Luận án tập trung vào giải pháp tối ưu truyền thông dựa trên kỹ thuật mã mạng.

1.2.2.3. Hệ phân tán triển khai trong máy chủ ảo

Theo tác giả Tanenbaum và Steen trong [124] định nghĩa hệ phân tán: *Hệ phân tán là tập các máy chủ độc lập cung cấp ứng dụng cho người dùng như là một hệ thống gắn bó duy nhất.*

Các tác giả Tanenbaum và Steen nhấn mạnh hai khía cạnh: Khía cạnh thứ nhất là một hệ thống phân tán bao gồm các thành phần phần cứng có thể tự trị. Khía cạnh thứ hai là người dùng (chương trình, phần mềm) nghĩ rằng họ đang làm việc với một hệ thống duy nhất.

Các thành phần tự trị cần phải hợp lực là trọng tâm trong việc phát triển các hệ thống phân tán. Về nguyên lý, ngay cả trong một hệ thống duy nhất có thể bao gồm từ các máy chủ hiệu năng cao đến các nút nhỏ trong các mạng cảm biến.



Hình 1.15: Hệ phân tán được tổ chức như phần mềm trung gian. Tầng phần mềm trung gian trải rộng trên nhiều máy chủ khác nhau và mỗi ứng dụng được cung cấp cùng giao diện tương tác [124]

Một đặc điểm quan trọng của hệ phân tán là sự khác biệt cách thức truyền thông giữa các máy chủ khác nhau và được ẩn khỏi người dùng. Một đặc điểm quan trọng

khác là người dùng và ứng dụng có thể tương tác với hệ phân tán một cách nhất quán và thống nhất mọi lúc, mọi nơi.

Về nguyên tắc, các hệ thống phân tán dễ mở rộng hoặc mở rộng quy mô lớn. Đặc điểm này là các máy chủ độc lập và ẩn cách thức các máy chủ này tham gia vào hệ thống đối với người dùng. Một hệ thống phân tán ở trạng thái sẵn sàng, liên tục trong khi một số máy chủ có thể tạm thời không hoạt động. Người dùng và ứng dụng đang thực thi không cần quan tâm đến các bộ phận đang được thay thế, sửa chữa hoặc thêm mới để phục vụ nhiều người dùng hơn, ...

Để hỗ trợ các máy chủ và mạng không đồng nhất, các hệ phân tán thường được tổ chức bằng một tầng phần mềm, nghĩa là được đặt một cách lô gic giữa tầng trên (người dùng, ứng dụng) và tầng dưới (các hệ điều hành, các phương tiện truyền thông cơ bản), như trong Hình 1.15. Theo đó, hệ phân tán như vậy có thể được gọi là phần mềm trung gian. Hình 1.15 cho thấy bốn máy chủ nối mạng và ba ứng dụng, trong đó ứng dụng B được phân tán trên các máy chủ 2 và 3. Mỗi ứng dụng được cung cấp cùng một giao diện tương tác. Hệ phân tán cung cấp phương tiện cho các thành phần của một ứng dụng phân tán duy nhất để truyền thông với nhau, nhưng cũng để truyền thông cho các ứng dụng khác nhau.

Theo các tác giả Coulouris, Jean và Tim trong [21] định nghĩa về hệ phân tán: *Hệ phân tán là một hệ thống trong đó các thành phần phần cứng hoặc phần mềm kết nối thông qua truyền thông mạng máy tính và hợp lực các hành động giữa các máy chủ thông qua truyền thông điệp. Định nghĩa đơn giản này bao gồm toàn bộ phạm vi của các hệ thống trong đó các máy chủ nối mạng có thể được triển khai một cách hữu ích.*

Các máy chủ được kết nối qua mạng có thể được phân chia theo bất kỳ khoảng cách nào. Hệ thống có thể ở các châu lục, trong cùng một tòa nhà hoặc trong cùng một phòng. Định nghĩa về các hệ phân tán có các đặc tính quan trọng sau:

Đồng thời: Trong một mạng máy chủ, thực thi chương trình đồng thời tuân theo tiêu chuẩn, quy tắc. Khả năng của hệ thống để xử lý các tài nguyên được chia sẻ có thể được tăng lên bằng cách thêm nhiều tài nguyên hơn (ví dụ: máy chủ) vào mạng. Sự hợp lực của các chương trình thực hiện đồng thời chia sẻ tài nguyên cũng là một đặc tính quan trọng.

Không có đồng hồ toàn cục: Khi các chương trình cần hợp lực, chúng phối hợp hành động bằng cách trao đổi thông điệp. Hợp lực phụ thuộc vào chia sẻ về thời gian xảy ra các hành động của chương trình. Nhưng nó chỉ ra rằng có giới hạn về độ chính xác mà các máy chủ trong mạng có thể đồng bộ hóa đồng hồ của chúng - không

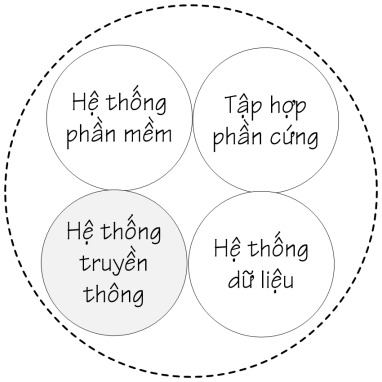
có khái niệm thời gian chính xác toàn cục duy nhất. Truyền thông duy nhất là bằng cách truyền thông điệp qua mạng.

Lỗi độc lập: Tất cả các hệ thống máy chủ có thể bị lỗi, và trách nhiệm của các nhà thiết kế hệ thống là lập kế hoạch sửa lỗi có thể xảy ra. Hệ thống phân tán có thể lỗi theo những cách mới. Lỗi trong mạng dẫn đến sự cô lập của các máy chủ được kết nối với nó, nhưng điều đó không có nghĩa là hệ phân tán dừng hoạt động. Trên thực tế, các chương trình trên chúng có thể không thể phát hiện xem mạng có bị lỗi hay trở nên chậm bất thường hay không. Tương tự, lỗi của máy chủ hoặc lỗi kết thúc chương trình trong hệ thống (sự cố), không được biết ngay lập tức với các thành phần khác mà nó giao tiếp. Mỗi thành phần của hệ thống có thể lỗi độc lập trong khi đó các thành phần khác vẫn chạy.

Mục đích chính để xây dựng và sử dụng các hệ phân tán bắt nguồn từ chia sẻ tài nguyên dùng chung. Thuật ngữ "tài nguyên" khá trừu tượng, nhưng đặc trưng nhất cho phạm vi của những thứ có thể được chia sẻ hữu ích trong một hệ thống mạng máy tính. Nó mở rộng từ các thành phần phần cứng như đĩa và máy in đến các thực thể được xác định bằng phần mềm như tệp, cơ sở dữ liệu và các đối tượng dữ liệu.

Hệ phân tán truyền thống trong [65] được tác giả Lê trình bày định nghĩa như sau: *Hệ tin học phân tán hay nói ngắn gọn là hệ phân tán (Distributed System) là hệ thống xử lý thông tin bao gồm nhiều bộ xử lý hoặc bộ vi xử lý nằm tại các vị trí khác nhau và được liên kết với nhau thông qua phương tiện viễn thông dưới sự điều khiển thống nhất của một hệ điều hành.*

Hình 1.16 thể hiện 4 thực thể của hệ phân tán, hệ thống này bao gồm các máy chủ kết nối qua hệ thống truyền thông dưới sự điều hành thống nhất của một hệ điều hành gọi là hệ điều hành phân tán.



Hình 1.16: Bốn thực thể của hệ tin học phân tán [65]

Dựa trên các định nghĩa về hệ phân tán theo [65, 124, 21], hệ phân tán hay còn gọi là chương trình phân tán của Luận án tập trung vào xây dựng hệ phân tán điều khiển cung cấp tài nguyên dùng chung. Tài nguyên dùng chung thể hiện trong Luận án là các thực thể được xác định bằng phần mềm, đó chính là cơ sở dữ liệu theo [21]. Bộ cung cấp tài nguyên trong hệ phân tán của Luận án tham chiếu đến các trường trong cơ sở dữ liệu để cung cấp tài nguyên dùng chung, người dùng thông qua giao diện tương tác để lựa chọn, đăng ký tài nguyên, đó chính là các trường dữ liệu trong cơ sở dữ liệu.

Các nghiên cứu về hệ phân tán tập trung vào các vấn đề như:

- Trật tự tổng quát chặt chẽ trình bày trong [14, 60, 65]
- Các vấn đề hợp lực trong hệ phân tán trình bày trong [10, 30]

Các thuật toán truyền thông nêu lên vấn đề truyền multicast trong hệ phân tán và giải pháp tối ưu truyền thông dựa trên kỹ thuật mã mạng. Bên cạnh đó, việc áp dụng kỹ thuật này cho hệ thống truyền thông trong Điện toán đám mây là xu hướng tất yếu được trình bày trong các nghiên cứu [35, 142].

Các ưu điểm của hệ phân tán so với hệ tập trung được Luận án trình bày qua các điểm chính sau:

- *Truy cập tài nguyên số lượng lớn* là điểm thuật lợi và quan trọng của hệ đối với người sử dụng. Trong các hệ tập trung, khi số lượng người dùng truy cập lớn dẫn đến tình trạng tắc nghẽn và hiệu năng xử lý của hệ giảm dẫn đến truy cập khó khăn đối với người dùng. Bên cạnh đó, nếu máy chủ trong hệ tập trung bị sự cố vì bất cứ nguyên nhân nào thì người dùng không thể truy cập được. Ngược lại, đối với hệ phân tán, khi số lượng người dùng truy cập vào một máy chủ lớn, một hệ thống cảnh báo tắc nghẽn sẽ chuyển truy cập người dùng sang máy chủ khác có khối lượng xử lý ít hơn để đảm bảo truy cập. Hệ thống cảnh báo này gọi là bộ phân phối tải.
- *Tính trong suốt phân tán* đối với người dùng là đặc điểm tiếp theo mà người sử dụng không cần biết mình đang truy cập ở máy chủ nào, các quá trình xử lý diễn ra bên trong hệ. Người dùng chỉ quan tâm là tài nguyên mình đăng ký truy cập có dễ dàng và có được chấp nhận hay không?
- *Tính gắn bó* là một trong những đặc điểm quan trọng của hệ, một tài nguyên dùng chung chỉ cung cấp duy nhất cho một người sử dụng và trạng thái này

phải giống nhau trên tất cả các máy chủ. Để đảm bảo tính gắn bó này, sự hợp lực giữa các máy chủ phải giải quyết được tương tranh, phòng tránh bế tắc và thống nhất giữa các máy chủ, đồng thời thực hiện trên cùng một giải thuật theo định nghĩa của hệ phân tán nêu trên.

- *Tính mở* là khả năng tăng các máy chủ xử lý khi số lượng truy cập của các máy chủ đã đạt tới hạn có khả năng xảy ra tắc nghẽn bởi các máy chủ không thể tự tăng tài nguyên của mình lên được nữa do giới hạn về khả năng phần cứng.

- *Tính chịu lỗi* là khi một hay nhiều máy chủ bị sự cố và vào lại sau khi đã khôi phục thì yêu cầu tính gắn bó phải đảm bảo cho toàn hệ. Hệ dừng khi tất cả các máy chủ trong hệ dừng hoạt động.

Dựa vào các đặc tính kỹ thuật nêu trên, để triển khai hệ cần thiết phải nghiên cứu cơ chế kỹ thuật đủ mạnh. Bộ cung cấp tài nguyên của hệ được xây dựng dựa trên tính gắn bó trong toàn hệ, trong đó các vấn đề được nêu khi nghiên cứu và triển khai hệ như: phòng chống bế tắc, loại trừ tương hỗ, đồng bộ hóa tiến trình, điều khiển tải,... đảm bảo tính gắn bó mạnh trong toàn hệ để cung cấp tài nguyên với thời gian thực. Nếu không đảm bảo các yếu tố trên dẫn đến hệ sẽ bị phá vỡ và không sử dụng được.

Trên cơ sở trình bày các đặc điểm của hệ phân tán, Luận án tập trung vào *giải pháp đảm bảo gắn bó dữ liệu và tối ưu truyền thông trong hệ thống ảo hóa.*

Theo công ty DIGISTAR [2] trình bày so sánh chi tiết Cloud Server và Máy chủ vật lý thể hiện qua Bảng 1.9. Bảng 1.9 thể hiện các tính năng nổi trội khi triển khai trên môi trường đám mây so với môi trường truyền thống. Các tính năng thể hiện: Cách thức hoạt động, Độ ổn định, Tính sẵn sàng, Khả năng mở rộng, Chi phí, Cấu hình lại Server và Sao chép Server. Để minh chứng tính hiệu quả triển khai hệ phân tán trên môi trường đám mây so với môi trường truyền thống, các nghiên cứu và triển khai thực tế trình bày qua những nội dung sau:

Nghiên cứu trong [26] của tác giả Divakarla và Kumari trình bày năm đặc tính thiết yếu của việc triển khai hệ phân tán trong môi trường đám mây: dịch vụ theo yêu cầu, truy cập mạng băng thông rộng, nhóm các tài nguyên, độ co giãn nhanh chóng và các dịch vụ đo được.

Nghiên cứu trong [74] của Laxmi và Krishnaiah xét kỹ thuật tính toán lưới là sự lựa chọn các máy tính từ nhiều nơi để thực hiện chia sẻ tài nguyên. Các tác giả xem

[2]https://www.digistar.vn/nen-su-dung-cloud-server-hay-may-chu-vat-ly, truy cập ngày 22/12/2019

lưới như là hệ phân tán thực hiện xử lý khối lượng lớn thông tin và lưới được là nhóm các nguồn tài nguyên để nhiều người sử dụng truy cập đồng thời. Ngoài ra, các tác giả cũng nêu các thuận lợi của tính toán lưới trong phần 3.2 của nghiên cứu [74] như:

- Tính toán lưới có thể giải quyết các vấn đề lớn, phức tạp hơn trong thời gian ngắn hơn.
- Dễ dàng cộng tác với các hệ thống khác.
- Sử dụng các phần cứng hiện có tốt hơn.
- Sử dụng các tài nguyên nhàn rỗi hiệu quả hơn.
- Môi trường lưới có nhiều mô-đun hơn và không có điểm lỗi cục bộ.
- Các tác vụ tự động khởi động lại nếu xảy ra lỗi.
- Các chính sách được quản lý bởi phần mềm lưới.
- Nâng cấp được thực hiện nhanh chóng mà không cần lên lịch ngừng hoạt động.
- Hiệu năng công việc được thực hiện xử lý song song.

Bên cạnh đó, nghiên cứu trong [88] của các tác giả Nerkar và Shinkar trình bày thiết kế và kiến trúc hệ thống của một số ứng dụng của Điện toán đám mây, trong đó nêu bật sử dụng lõi là kỹ thuật hệ phân tán.

Qua Bảng 1.9 cho thấy, hiệu năng cao của việc triển khai hệ phân tán trên máy ảo thể hiện qua các tính năng: Độ ổn định, Tính sẵn sàng và Khả năng mở rộng. Trong thực nghiệm của Luận án, khi triển khai trên máy ảo theo Hình 1.17. Hệ phân tán được triển khai trên máy ảo có hệ điều hành là Windows Server 2012. Trong trường hợp nếu máy chủ vật lý bị sự cố về phần cứng thì máy ảo thực thi hệ phân tán vẫn hoạt động bình thường hoặc di trú sang hệ thống máy chủ ảo khác trong hệ thống đã được cấu hình. Theo Hình 1.17, hệ thống thực nghiệm có 3 Cluster được thiết lập trên 3 hệ thống máy chủ vật lý. Các hệ thống máy ảo cấu hình dưới dạng Cluster: Cluster1 và Cluster3 bao gồm 4 máy chủ vật lý, Cluster2 bao gồm 3 máy chủ vật lý. Khi 1 máy chủ vật lý bị sự cố hoặc tắt, các máy ảo phía bên trên lập tức di trú sang máy chủ vật lý khác trong Cluster hoặc các cluster khác mà không gây gián đoạn hoạt động. Do đó, tính sẵn sàng của máy ảo này rất cao để phục vụ các kết nối từ phía người dùng. Người dùng không bị gián đoạn truy cập khi lỗi phần cứng của máy chủ vật lý vì bất cứ nguyên nhân nào.

Đối với chi phí khi triển khai hệ phân tán trên máy ảo, có rất nhiều cách so sánh khác nhau khi tính đến chi phí như: đầu tư phần cứng máy chủ, bản quyền phần mềm, hạ tầng hỗ trợ, người vận hành, chế độ bảo dưỡng, bảo trì, ... Trong Luận án chỉ đưa ra phương pháp so sánh mức đầu tư chung, không tính chi phí sửa chữa hoặc các tiện ích khác để hoạt động hệ phân tán trên các máy chủ đó. Theo Blog của ServerMania[3], việc tính toán chi phí cho 5 năm vận hành và sử dụng sẽ tiết kiệm 74 phần trăm chi phí giữa việc sử dụng máy chủ đám mây so với sử dụng máy chủ đầu tư trực tiếp. Theo Blog của SherWeb[4], việc tính toán chi phí cho 5 năm vận hành và sử dụng sẽ tiết kiệm 79 phần trăm chi phí giữa việc sử dụng máy chủ đám mây so với sử dụng máy chủ đầu tư trực tiếp.

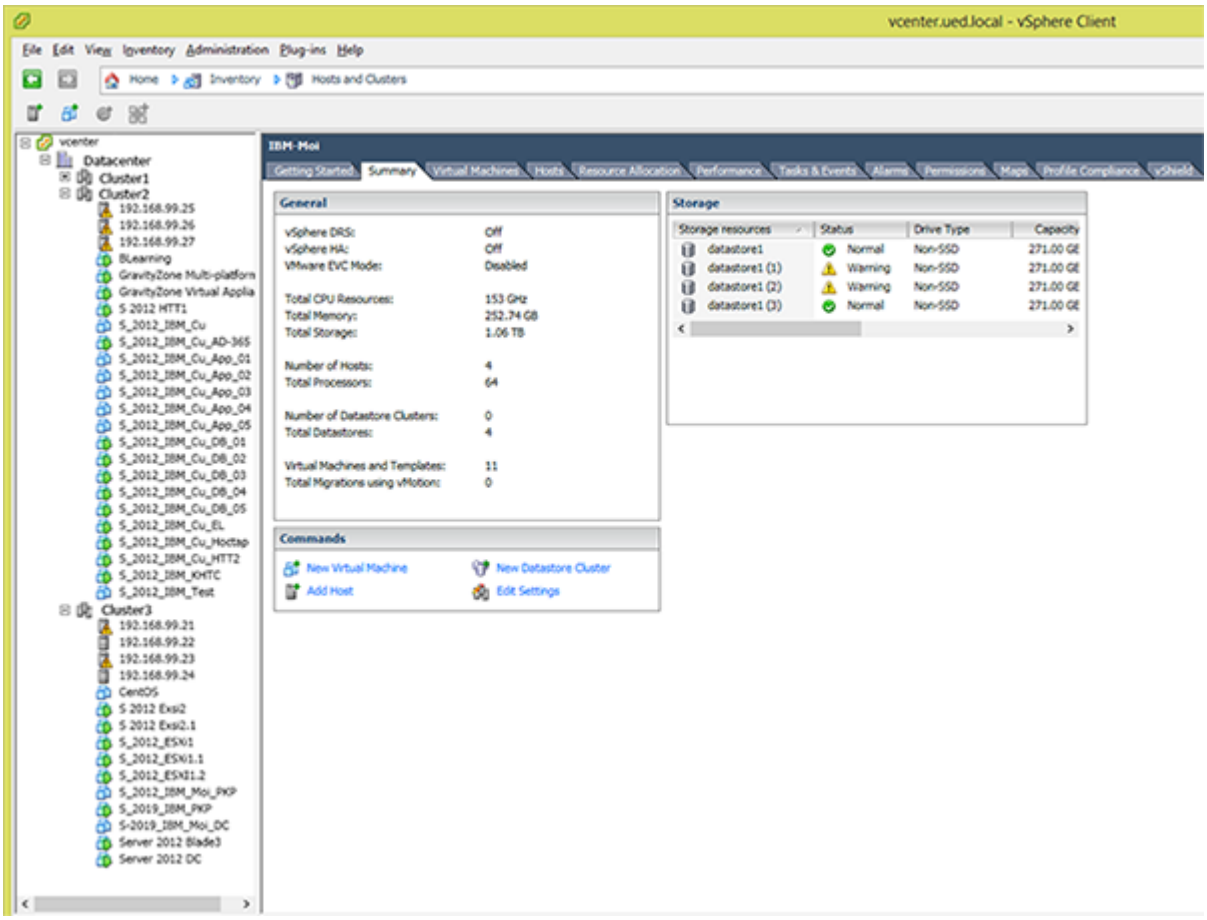
Từ các công trình nghiên cứu [26, 74, 88] và triển khai thực nghiệm có thể tóm lược: *hệ phân tán trong máy ảo hoạt động hiệu năng cao và chi phí thấp so với triển khai trên môi trường truyền thống.*

Nghiên cứu trong [37] của các tác giả Galante và Bona trình bày cơ chế co giãn của các hệ thống đám mây trong thương mại và nghiên cứu. Các tác giả cũng đã trình bày tính co giãn là đặc tính chính của Điện toán đám mây, đây là thành phần phân biệt so với các hệ thống hiện có như cụm máy chủ và tính toán lưới. Độ co giãn đã thay đổi cách thiết kế và xây dựng ứng dụng môi trường máy tính. Theo các tác giả Boutaba, Zhang và Zhani trong Chương 17 [7] trình bày các lợi ích, thách thức và tiếp cận trong môi trường đám mây. Trong nghiên cứu, các tác giả đã trình bày sự phù hợp với từng loại đám mây trong quá trình di trú với các chi phí truyền thông và độ trễ. Trong quá trình di trú, thời gian chết phụ thuộc vào từng loại kiến trúc và nền tảng ảo hóa.

Trong thực nghiệm của Luận án, cấu hình một máy ảo thực thi hệ điều hành thể hiện qua Hình 1.18. Cấu hình này là cấu hình lô gic, nghĩa là, các tham số thành phần như CPU, RAM, HDD,... có thể tùy chỉnh phụ thuộc vào hiệu năng đang sử dụng. Các cấu hình lô gic này không phụ thuộc vào phần cứng vật lý. Khả năng cung cấp tối đa cho các tham số thành phần là giá trị nhàn rỗi còn lại của hệ thống ảo hóa, ví dụ theo Hình 1.18, khả năng tăng tối đa RAM cho máy ảo là 4080 GB. Trong trường

[3]https://blog.servermania.com/iaas-vs-on-premise-server-cost-comparison, truy cập ngày 22/11/2019

[4]https://www.sherweb.com/blog/cloud-server/total-cost-of-ownership-of-servers-iaas-vs-on-premise, truy cập ngày 22/11/2019



Hình 1.17: Hệ thống ảo hóa VMWare

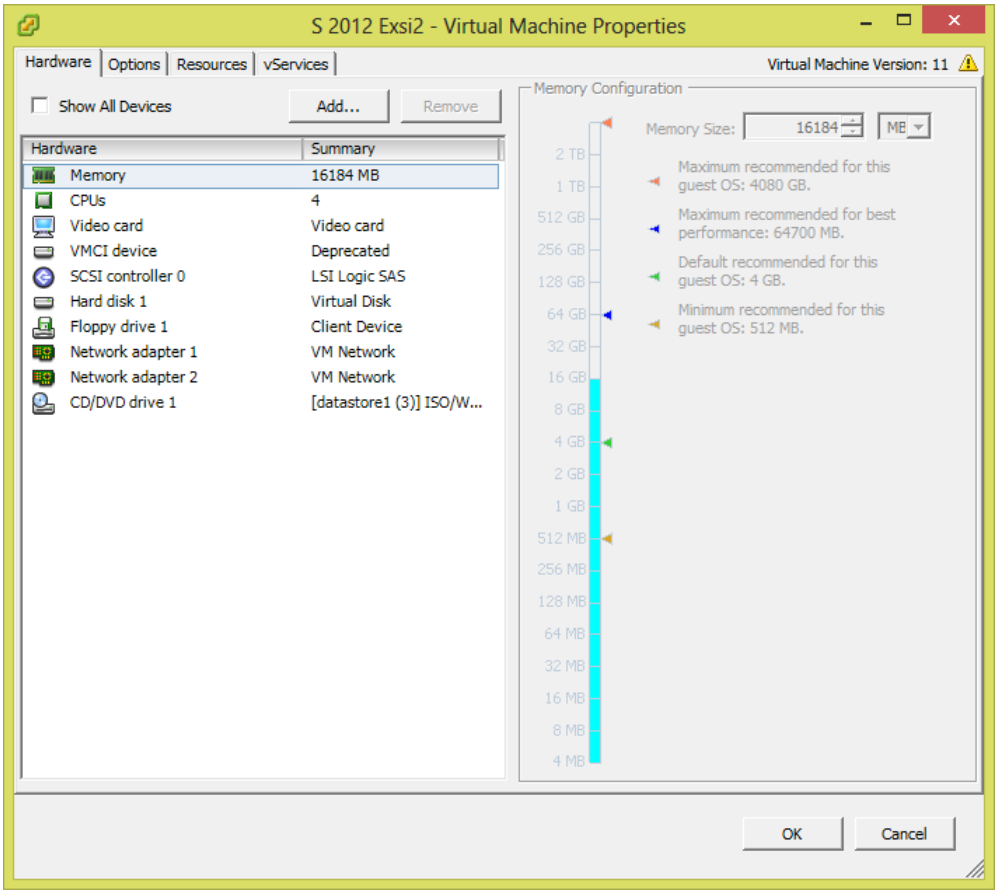
Bảng 1.9: So sánh chi tiết Cloud Server và Máy chủ vật lý

Tính năng	**Máy chủ vật lý**	**Cloud Server/Cloud VPS**
Cách thức hoạt động	Là một máy chủ vật lý	Lưu trữ và hoạt động trên hệ thốnghạ tầng Cloud (điện toán đám mây)
Độ ổn định	Khi máy chủ vật lý hỏng thì toàn bộ hệ thống ngừng hoạt động Ổ cứng hỏng sẽ gây mất dữ liệu, dường như không thể khôi phục	Công nghệ Cloud giúp hệ thống hoạt động ổn định, khả năng uptime 99.99 phần trăm Tất cả các thành phần đều được thiết lập dự phòng, và tự đông thay thế khi bị hỏng đảm bảo hoạt động bình thường nên hệ thống thông tin của bạn luôn an toàn và có tính sẵn sàng 24/7
Tính sẵn sàng	Tất cả data sẽ được lưu trữ trên máy vật lý (Rủi ro cao) Chỉ cần 1 HDD bị hỏng có thể toàn bộ data của bạn sẽ mất Việc triển khai backup tốn nhiều chi phí	Data lưu trữ tập trung trên hệ thống SAN không lưu trên Server vật lý Data được backup đều đặn Nếu có 1 Server vật lý bị lỗi, Cloud Server của bạn vẫn hoạt động ổn định
Khả năng mở rộng	Nâng cấp phức tạp vì phải mua thiết bị phần cứng chuyên dụng Thời gian downtime khi cần nâng cấp khá lâu	Có thể mở rộng hoặc thu hẹp ngay lập tức khi doanh nghiệp có yêu cầu Có thể mở rộng và thu hẹp nguồn tài nguyên không giới hạn dung lượng
Chi phí	Tốn khá nhiều chi phí đầu tư cho toàn bộ phần cứng Chi phí vận hành, bảo trì hệ thống cũng rất đáng kể. Chi phí triển khai backup (sao lưu) cao	Chỉ phải chi trả cho nhu cầu thực tế sử dụng (CPU, RAM, HDD, ... được cấu hình theo yêu cầu) Chi phí có thể linh hoạt thay đổi theo yêu cầu của doanh nghiệp
Cấu hình lại Server	Bạn phải mua thêm các thiết bị như RAM, HDD trong trường hợp nâng cấp và việc gắn thêm có thể gây ra nhiều rủi ro không mong muốn	Bạn có thể thiết lập lại cấu hình cho server 1 cách dễ dàng trong khoảng thời gian ngắn
Sao chép Server	Phức tạp vì server vật lý khó mà đáp ứng được việc này	Đơn giản, nhanh chóng

hợp cần tăng số lượng máy ảo thực thi hệ phân tán để đáp ứng yêu cầu người dùng, chúng ta có thể nhân bản bằng cách thực thi Clone. Clone là cách thức nhân bản một máy ảo nào đó được tạo ra trong VMware thành nhiều bản nữa giống hệt máy ảo ban đầu với mục đích tiết kiệm thời gian cho việc lặp đi lặp lại cài đặt các máy ảo có cấu hình giống nhau.

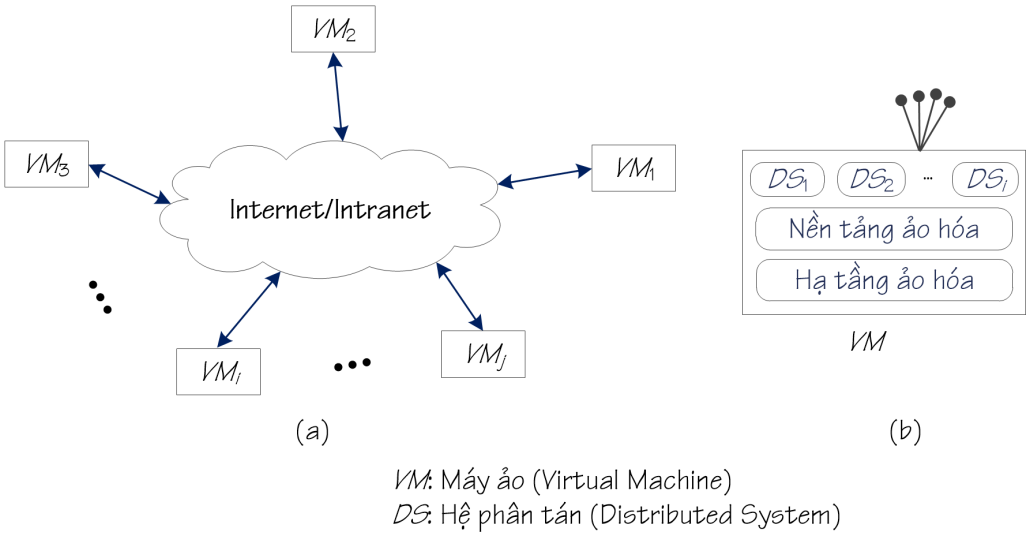
Do đó, *hệ phân tán trong máy ảo không phụ thuộc vào phần cứng nên có khả năng co giãn về mặt số lượng của hệ và có thể di trú sang hệ thống ảo hóa khác mà không mất kết nối.*

Theo tác giả Khan trong [55] trình bày Diện toán đám mây là một công nghệ mới,



Hình 1.18: Cấu hình của một máy ảo thực thi hệ phân tán

là sự mở rộng lớn của hệ phân tán. Vì vậy với cơ chế dịch vụ theo nhu cầu sử dụng, truy cập băng thông rộng và tự động mở rộng hoặc giảm thiểu số lượng các tài nguyên máy tính được phân phối cho ứng dụng vào bất kỳ thời điểm nào theo nhu cầu sử dụng (auto scaling) trong môi trường đám mây cho phép số lượng người dùng truy cập bất cứ thời điểm nào mà không gây tình trạng tắc nghẽn cục bộ so với môi trường truyền thống (cụm máy chủ và tính toán lưới). Theo BuycPanel[5] trình bày: Tự động mở rộng quy mô trong điện toán đám mây là chế độ trong đó các tài nguyên tính toán được phân bổ được tự động tăng hoặc giảm dựa trên nhu cầu của khách hàng trong bất kỳ giai đoạn nào. Điện toán đám mây là một cuộc cách mạng nổi bật trong việc cung cấp tài nguyên máy tính. Nó có thể xây dựng một thiết lập đám mây toàn diện và có thể mở rộng. Khi ứng dụng yêu cầu khả năng tính toán bổ sung, ta chỉ cần khởi chạy thêm tài nguyên tính toán theo yêu cầu; khi các tài nguyên tính toán không còn cần thiết nữa, hệ thống có thể thu hồi lại tài nguyên. Hệ phân tán trong máy ảo có khả năng hoặc động hiệu năng cao và co giãn về mặt số lượng để đáp ứng nhu cầu người dùng. Do đó, *hệ phân tán trong đám mây đáp ứng số lượng truy cập cực đại từ người dùng từ môi trường đám mây lớn hơn so với môi trường truyền thống.*



Hình 1.19: Hệ phân tán triển khai trên hệ thống máy ảo

Hiện nay, các ứng dụng đám mây xem như là các dịch vụ đối với người dùng triển khai ở 3 tầng. Thông qua các dịch vụ này cho phép hệ phân tán triển khai trên các VM tập trung tầng 3 của hệ thống ảo hóa theo Hình 1.19. Mỗi VM có thể triển khai một hoặc nhiều hệ phân tán theo Hình 1.19.b. Hệ phân tán cung cấp dịch vụ tài nguyên dùng chung cho người sử dụng tại các thiết bị đầu cuối, do đó, hệ phân tán triển khai

[5]https://www.buycpanel.com/choose-auto-scaling/, truy cập ngày 22/11/2019

tại tầng 3 của môi trường đám mây (SaaS). Do đó, hệ phân tán theo hướng nghiên cứu của Luận án là chương trình, phần mềm trung gian triển khai trên hệ điều hành và không triển khai trên Hypervisor. Theo mô tả trong Hình 1.2, Hình 1.15 và Hình 1.19, phạm vi của "*hệ phân tán trong máy ảo*" của Luận án là các chương trình phân tán, tính hỗ tạp của tài nguyên là phần mềm trung gian nhằm cung cấp tài nguyên dùng chung được triển khai trong hệ thống máy ảo.

Dựa trên đặc tính hệ phân tán trong máy ảo, Luận án đã thực hiện mô phỏng thuật toán 4PCoDT trình bày trong Mục 2.3.2. Chương trình phân tán được xây dựng trên ngôn ngữ Java, hệ quản trị cơ sở dữ liệu MySQL thực thi trên hệ điều hành Windows Server 2012 nằm trong hệ thống ảo hóa VMware vSphere. Khi triển khai trên môi trường ảo hóa, Luận án thực nghiệm với kịch bản:

1. Triển khai 5 máy ảo với chương trình phân tán được cài đặt trên hệ điều hành Windows Server 2008, mỗi chương trình phân tán có cơ sở dữ liệu cục bộ và bộ điều khiển cung cấp tài nguyên dùng chung để các client truy cập. Khi yêu cầu tài nguyên như CPU ảo, RAM ảo, băng thông ảo vượt quá nhu cầu sử dụng, chương trình phân tán vẫn hoạt động nhờ cơ chế tự nới rộng khi vẫn còn tài nguyên nhàn rỗi. Đối với môi trường truyền thống, khi tài nguyên vật lý (RAM, CPU, HDD,...) tới hạn sẽ làm chậm hoặc treo các tiến trình hoạt động, việc mở rộng tài nguyên vật lý phải chờ lắp đặt thêm và cấu hình lại.

2. Khi tắt một máy chủ vật lý trong hệ thống ảo hóa, lập tức các máy ảo chứa chương trình phân tán sẽ di trú sang các máy chủ vật lý khác trong khi các chương trình phân tán vẫn hoạt động cung cấp tài nguyên dùng chung cho người dùng.

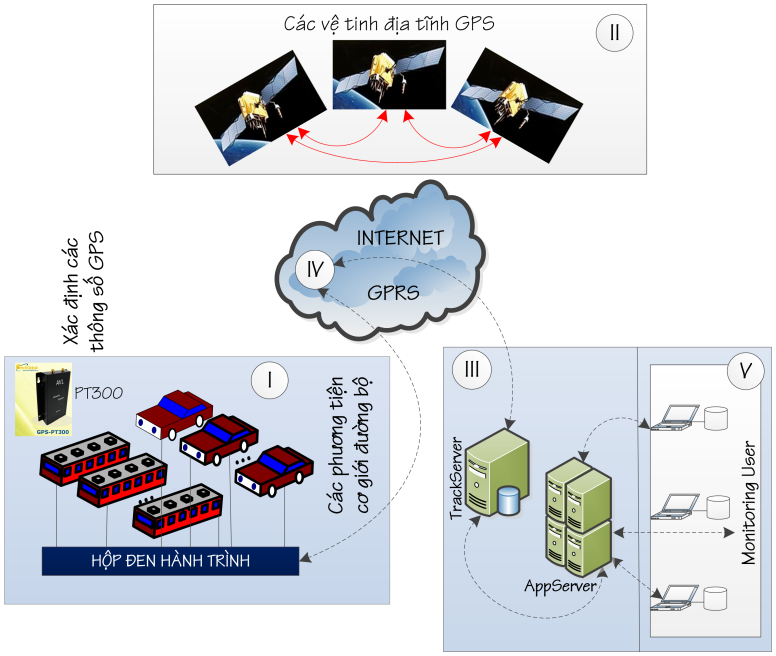
3. Khi cần tăng thêm số lượng máy chủ để đáp ứng cung cấp tài nguyên dùng chung khi số lượng truy cập lớn, tại máy ảo đang chứa chương trình phân tán được nhân bản (clone) thêm máy ảo mới mà không cần phải trang bị thêm máy chủ vật lý. Máy ảo mới sau khi nhân bản sẽ cấu hình lại tham số trên chương trình phân tán và vận hành ngay, không mất nhiều thời gian để cài đặt và cấu hình so với cài trên máy chủ vật lý trước đây.

1.3. Mô hình và giải pháp điều khiển cung cấp tài nguyên trong hệ thống máy chủ ảo

1.3.1. Giới thiệu bài toán

Giải pháp điều khiển cung cấp tài nguyên truyền thông cho hệ phân tán triển khai trong máy ảo là bài toán cho các ứng dụng lớn trong đám mây nhằm cho phép truyền tải thông tin từ nguồn đến tập đích một cách nhanh chóng với thông lượng cực đại. Những lĩnh vực áp dụng của nghiên cứu chủ yếu là các hệ thống đăng ký, giám sát tài nguyên trực tuyến như: y tế, giáo dục, giao thông, kinh doanh từ xa, ...

Ví dụ đầu tiên của hệ thống giám sát trực tuyến các phương tiện cơ giới đường bộ, sơ đồ tổng quát của hệ thống thể hiện qua Hình 1.20.



Hình 1.20: Sơ đồ tổng quát về hệ thống giám sát phương tiện cơ giới đường bộ

Thông qua các hoạt động mô tả trong Hình 1.20, các thiết bị theo dõi hành trình gửi dữ liệu về hệ thống Server giám sát để lưu trữ và kiểm soát trong suốt quá trình vận hành của phương tiện. Khoảng 5 đến 10 giây, thiết bị gửi thông điệp về cho hệ thống Server để báo tình trạng đang hoạt động của phương tiện.

Hình 1.21 mô tả kết nối truyền thông trong nước của các nhà cung cấp dịch vụ thể hiện kết nối các tỉnh thành về trung tâm cung cấp dịch vụ viễn thông tại 3 nơi: Hà Nội, Đà Nẵng và Hồ Chí Minh. Trong trường hợp triển khai hệ thống giám sát phương tiện cơ giới đường bộ, mỗi tỉnh/thành có một hệ thống máy ảo để chứa các máy chủ TrackServer và AppServer. Các máy chủ TrackServer và AppServer có nhiệm vụ giám sát, lưu trữ các gói tin từ các phương tiện cơ giới gửi về. Như vậy, một phương tiện

cơ giới di chuyển từ đầu đến cuối đất nước sẽ gửi dữ liệu gói tin về các máy chủ trong xuốt hành trình của mình.

Hình 1.21: Kết nối truyền thông trong nước của các nhà cung cấp dịch vụ

Đơn vị giám sát phương tiện cơ giới (Monitoring User) là người dùng đầu cuối, bao gồm: Doanh nghiệp vận tải, Công an giao thông, Thanh tra Sở Giao thông Công chính, Ủy ban An toàn giao thông, Bộ Giao thông vận tải, ... Monitoring User muốn truy xuất hành trình thì phải truy xuất dữ liệu tại tất cả các máy chủ giám sát mà phương tiện đã đi qua. Để thể hiện được thông tin, tại máy chủ mà đơn vị giám sát phương tiện cơ giới truy cập phải yêu cầu truy vấn đến máy chủ giám sát để lấy thông tin hành trình. Ngoài ra, Monitoring User có thể giám sát đồng thời nhiều phương tiện đang hoạt động mà mình đang quản lý. Vì vậy, lượng thông tin di chuyển giữa các máy chủ là rất lớn.

Như vậy, để kiểm soát quá trình hoạt động của phương tiện đã chiếm tài nguyên truyền thông lớn chưa kể đến việc trích xuất thông tin cả quá trình di chuyển. Do đó,

việc đảm bảo tính tối ưu truyền thông cho các thông điệp trong hệ thống là vấn đề cần quan tâm, xem xét. Hệ thống giám sát trực tuyến các phương tiện cơ giới đường bộ là một hệ thống lớn về quy mô và phức tạp về mặt kỹ thuật, vận hành và có quan hệ mật thiết với nhiều hệ thống lớn khác.

Trên cơ sở đưa ra các kết luận về hệ thống này, thách thức đưa ra trong nghiên cứu về mặt kỹ thuật đó là phương pháp và công nghệ phân tán như: đồng bộ giữa các tiến trình đang hoạt động ở xa, xử lý song song, vấn đề tắc nghẽn đường truyền, thời gian lô gíc, sự nhất quán và gắn bó thông tin,... cần phải được tiếp tục quan tâm, nghiên cứu và phát triển. Dựa vào đó, Luận án đã trình bày, đề xuất các hướng tiếp theo của nghiên cứu cho ví dụ:

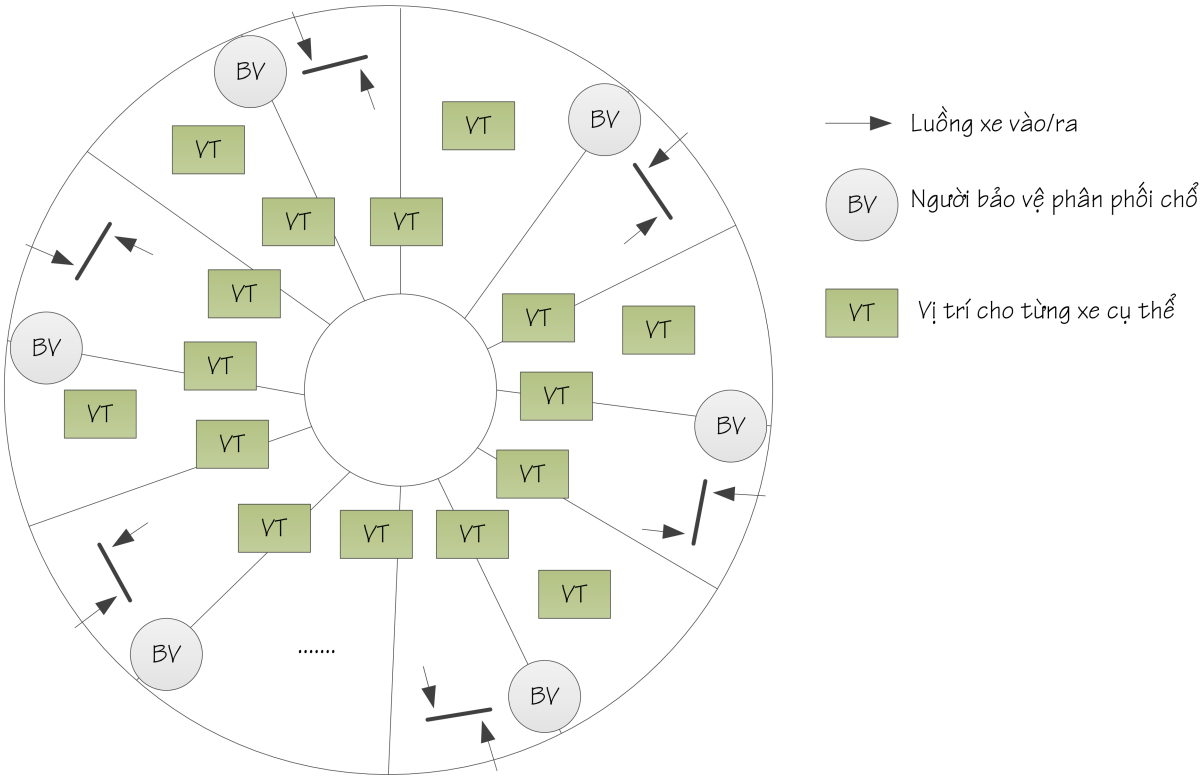
1. Triển khai hệ thống giám sát trên nền tảng phân tán trong môi trường đám mây để đáp ứng khả năng truy cập với số lượng tiến trình lớn và có khả năng mở rộng không gian lưu trữ.
2. Xây dựng giải pháp tối ưu truyền thông giữa các hệ thống giám sát dựa trên kỹ thuật mã mạng.

Một số ứng dụng đối với bài toán đăng ký tài nguyên trực tuyến theo Hình 1.1 như: hệ thống thương mại điện tử, đăng ký tín chỉ tại các trường đại học Việt Nam,... Trong quá trình đăng ký, các thông điệp trao đổi giữa người sử dụng và Server rất lớn trong khi thông tin cần hiển thị đối với người sử dụng là thông tin thực để người sử dụng có sự lựa chọn đúng.

Một ví dụ thứ hai lấy từ ví dụ trong Chương IV của tác giả Lê [65]. Hình 1.22 trình bày mô phỏng bài toán bãi đỗ xe, trong đó:

1. BV: người bảo vệ có nhiệm vụ phân phối chổ cho các xe ô tô. BV chính là hệ thống điều khiển cung cấp tài nguyên trong hệ phân tán. Trong ví dụ có 5 BV tương ứng 5 chương trình được cài đặt trên máy chủ.
2. VT: vị trí cho từng xe ô tô cụ thể. VT là tài nguyên dùng chung đối với người sử dụng trong hệ phân tán. Vị trí chính là tài nguyên cung cấp cho người sử dụng.
3. Các mũi tên: mô tả luồng vào/ra của ô tô. Mũi tên là các tiến trình yêu cầu tài nguyên (VT) trong hệ phân tán.

Phát biểu bài toán như sau:



Hình 1.22: Mô phỏng bài toán bãi đỗ xe [65]

Tình huống thứ 1: Giả sử rằng bãi để xe ô tô là loại bãi lớn có một cổng vào dưới sự kiểm soát của một người bảo vệ (BV) duy nhất. BV chỉ biết được một phần của trạng thái bãi để xe. Trong khi BV nghĩ rằng bãi xe đã bị đầy, thì khi đó lại có nhiều lái xe đang cho xe chạy ra cổng. Trong trường hợp này, BV không giải quyết cho các xe khác tiếp tục vào bãi nữa, mặc dù lúc này trong bãi đang có chổ trống. Như vậy, BV không nắm được trạng thái hiện hành của bãi.

Tình huống thứ 2: Nếu ta có bãi để xe với nhiều cổng vào và tại mỗi cổng vào có BV, thì mỗi BV chỉ có thể được biết trạng thái với độ trễ nhất định và điều đó dẫn đến tình huống thứ 2. Đó là tình huống có nhiều trung tâm ra quyết định như trong Hình 1.22.

Trên thực tế, BV nào đó tin rằng không còn chổ trống nữa, trong khi BV khác lại vừa mới cho ra khỏi bãi một số xe mà BV chưa kịp báo cho các BV khác. Cũng có thể diễn ra trường hợp là cùng một lúc các BV giải quyết các xe vào cùng một vị trí trong bãi do vì họ thiếu thông tin.

Xét trường hợp xảy ra không gắn bó. Một người bảo vệ có vai trò như là chương trình cung cấp chổ của bãi để xe ô tô. Ở đây, chổ để xe được xem như là tài nguyên của hệ. Giả sử rằng ở thời diểm cho trước ta có 4 người bảo vệ và có 100 chổ còn

trống. Tất cả các người bảo vệ đều có thông tin đó. Trạng thái của hệ lúc này là *gắn bó*. Ba trong số họ phát đi các thông tin trong Bảng 1.10 như sau:

Bảng 1.10: Thông điệp phát đi trong hệ thống

Ký hiệu	**Thông tin phát đi**
$M1$	Thêm 20 chổ trống
$M2$	Đã có 10 chổ bị chiếm
$M3$	Dành 10% chổ trống để quét dọn sân bãi

Bảng 1.11 thể hiện kết quả khi không có ràng buộc nào đối với trình tự xử lý các thông điệp nhận được của các người bảo vệ thì sẽ có số lượng chổ trống khác nhau. Điều này dẫn đến phá vỡ nguyên lý hoạt động chung của hệ thống. Để bảo đảm các bản cập nhật giống nhau thì trình tự cập nhật nhất thiết phải giống nhau trên tất cả các trạm.

Bảng 1.11: Sự không gắn bó giữa bốn người bảo vệ

Trật tự xử lý	**Bảo vệ 1**		**Bảo vệ 2**		**Bảo vệ 3**		**Bảo vệ 4**	
	Thông điệp	Giá trị	Thông điệp	Giá trị	Thông điệp	Giá trị	Thông điệp	Giá trị
		100		100		100		100
1	$M1$	**120**	$M1$	**120**	$M3$	**90**	$M2$	**90**
2	$M3$	**108**	$M2$	**110**	$M1$	**110**	$M3$	**81**
3	$M2$	**98**	$M3$	**99**	$M2$	**100**	$M1$	**101**

Như vậy, các người bảo vệ phải hợp lực với nhau để phân phối chính xác các chổ trong bãi. Đặc biệt là số lượng chổ còn trống càng ít thì vai trò của hợp lực càng quan trọng.

Qua hai ví dụ điển hình đối với ứng dụng phân tán, các yêu cầu kỹ thuật trong hệ phân tán cần thiết phải được xây dựng để đáp ứng số lượng lớn các tiến trình và đảm bảo tính gắn bó diễn ra trong hệ phân tán.

1.3.2. Mô hình tổng quát

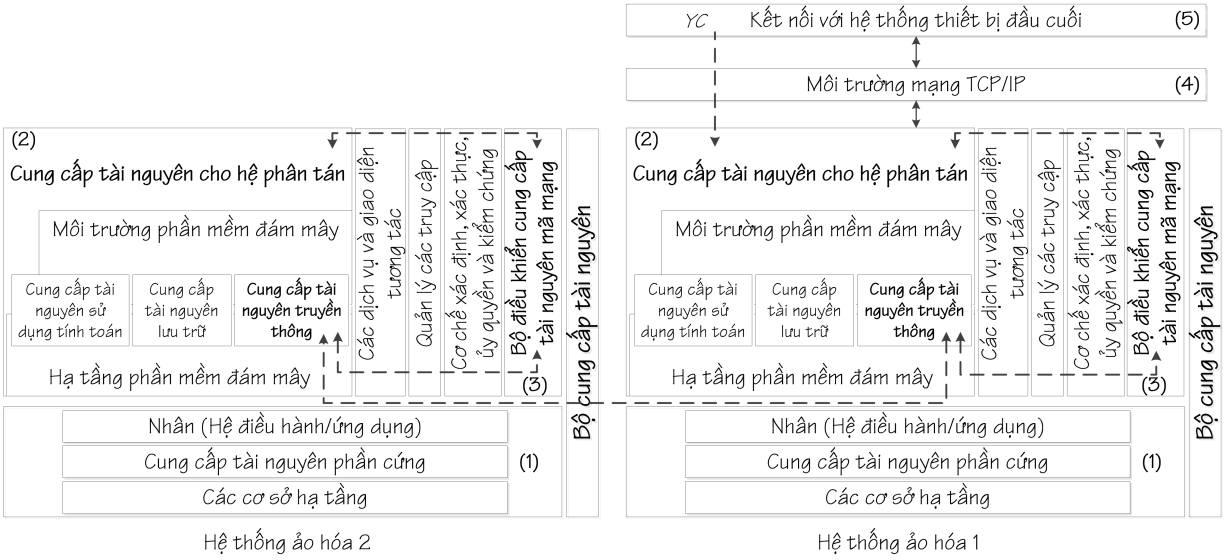
Dựa vào mô hình cung cấp tài nguyên trong đám mây của tác giả Goncalves trong nghiên cứu [39] và các vấn đề cơ sở về cung cấp tài nguyên trong hệ thống máy ảo, mô hình tổng quát Luận án đề xuất được trình bày theo Hình 1.23 mô tả các bộ điều khiển cung cấp tài nguyên. Mô hình bao gồm 5 phần cơ bản:

- *Phần 1* là cung cấp tài nguyên vật lý.
- *Phần 2* cung cấp tài nguyên ảo hóa hay còn gọi cung cấp dịch vụ đám mây.
- *Phần 3* là bộ điều khiển cung cấp tài nguyên ảo hóa.
- *Phần 4* là môi trường truyền thông kết nối với các đám mây khác hoặc người sử dụng.
- *Phần 5* là các thiết bị đầu cuối kết nối vào đám mây để yêu cầu tài nguyên.

Trong *phần 1* của mô hình thể hiện việc cung cấp tài nguyên vật lý phục vụ các ứng dụng trong hệ thống ảo hóa, nguồn tài nguyên này là tổng từng thành phần vật lý được kết nối. Ràng buộc của tài nguyên là không vượt quá giới hạn vật lý cho phép, điều này có nghĩa yêu cầu tài nguyên của hệ thống ảo hóa vượt quá nhu cầu cung cấp vật lý. Trong trường hợp tài nguyên vật lý tới hạn, việc mở rộng bằng cách thêm các thiết bị cùng loại để đảm bảo quá trình cung cấp. Trong nghiên cứu, Luận án tập trung giải quyết các vấn đề *tương tranh, bế tắc* đối với cung cấp tài nguyên dùng chung.

Phần 2 thể hiện hệ thống ảo hóa, hệ thống này đảm nhiệm vai trò cung cấp tài nguyên như là dịch vụ, đây là thành phần chính để được công nhận là đám mây. Các dịch vụ đối với người sử dụng là tài nguyên ảo, quá trình cung cấp thông qua chiến lược để đảm bảo tối ưu đối với tài nguyên. Dịch vụ trong phần này phải được cung cấp, giám sát và thu hồi một cách tự động. Các dịch vụ ảo hóa là vấn đề được quan tâm bởi các nhà nghiên cứu, triển khai và cũng là đầu vào cho các thuật toán tối ưu. Quá trình cung cấp tài nguyên dùng chung đối với người sử dụng đảm bảo các đặc tính của hệ phân tán, Luận án giải quyết tính gắn bó trong hệ thông qua giải pháp *cung cấp tài nguyên cho hệ phân tán*. Bên cạnh đó, Luận án nghiên cứu giải pháp *cung cấp tài nguyên truyền thông* để đảm bảo hoạt động truyền thông điệp của hệ phân tán trong môi trường ảo hóa.

Phần 3 mô tả bộ điều khiển cung cấp tài nguyên ảo hóa đảm nhiệm vai trò cung cấp, giám sát toàn bộ các hành vi tài nguyên hiện có trong hệ thống. Thành phần tập



Hình 1.23: Mô hình tổng quát cung cấp tài nguyên trong hệ thống máy ảo

trung nghiên cứu của Luận án là *bộ điều khiển cung cấp tài nguyên truyền thông mã mạng* đóng vai trò tối ưu băng thông, tránh trùng lặp thông tin đồng thời đạt thông lượng cực đại tại tập đích.

Môi trường truyền thông trong *phần 4* là thành phần quan trọng của đám mây cho phép người sử dụng hoặc các đám mây khác kết nối nhằm chia sẻ tài nguyên dùng chung. Môi trường truyền thông vốn có độ trễ và giới hạn băng thông, do đó, cung cấp tài nguyên với thời gian nhanh nhất là một trong những bài toán cần quan tâm nghiên cứu. Luận án nghiên cứu giải pháp *kỹ thuật mã mạng* để giải quyết bài toán tối ưu trong truyền thông.

Thiết bị đầu cuối trong *phần 5* là tập hợp tất cả các thiết bị cho phép kết nối qua môi trường truyền thông để yêu cầu tài nguyên trong đám mây. Các thiết bị đầu cuối có thể là các đám mây khác cộng tác hoặc người sử dụng đăng ký tài nguyên dùng chung của hệ thống. Các thiết bị đầu cuối không quan tâm đến quá trình hoạt động diễn ra trong máy ảo mà chỉ quan tâm đến tài nguyên có được cấp phát hay không và thời gian đáp ứng tài nguyên.

Việc mô tả các thành phần cơ bản trong Hình 1.23 đặt ra bài toán tổng quát quá trình cung cấp tài nguyên đám mây như sau:

- Các thiết bị đầu cuối trong phần 5 yêu cầu n tài nguyên dùng chung $\mathbb{YC} = \sum_{i=1}^{n} yc_i$ đến bộ cung cấp tài nguyên trong hệ phân tán trong phần 2 thông qua môi trường truyền thông trong phần 4.

- Sau khi tiếp nhận và xử lý, $\mathbb{YC}$ gửi đến bộ cung cấp tài nguyên mã mạng trong phần 3 để xử lý.
- Bộ cung cấp tài nguyên trong phần 3 chia $\mathbb{YC}$ thành các véc tơ, mã hóa và truyền đến các máy chủ trong hệ phân tán thông qua bộ cung cấp tài nguyên truyền thông trong phần 2.
- Các véc tơ được truyền đến tập đích là các S_i trong hệ phân tán tiếp nhận, phân tích, tính toán, xử lý và hồi đáp tài nguyên.
- Sau khi hoàn thành việc xử lý tài nguyên dùng chung, bộ cung cấp tài nguyên truyền thông mã mạng trả kết quả đến thiết bị đầu cuối trạng thái tài nguyên dùng chung.

Bài toán tổng quát vừa nêu đặt ra hai yêu cầu cơ bản đó là *đảm bảo yêu cầu cung cấp tài nguyên dùng chung* và *tối ưu truyền thông* giữa các thành phần tham gia vào hệ thống.

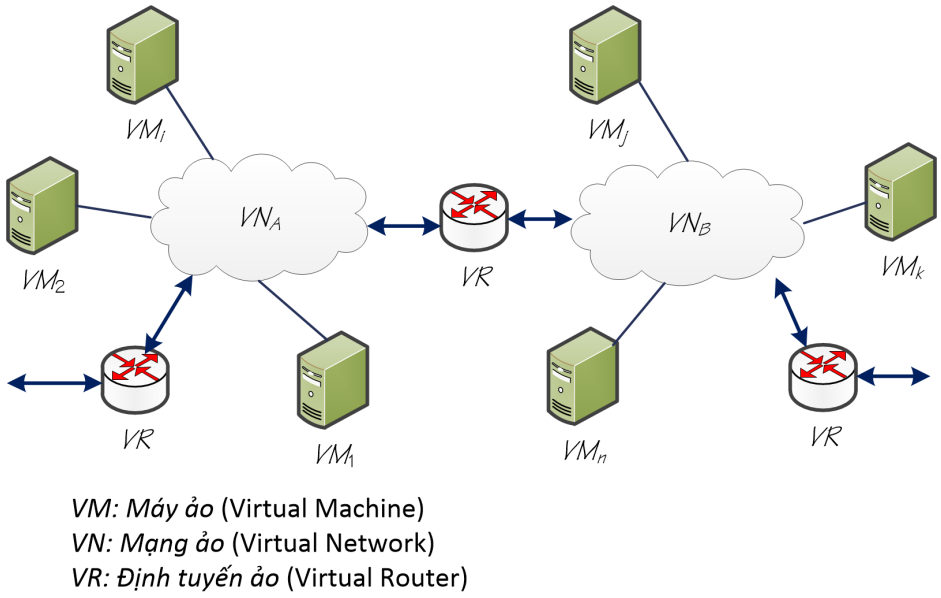
Bài toán tổng quát của Luận án có thể phát biểu như sau: *Hệ thống điều khiển cung cấp tài nguyên cho hệ phân tán triển khai trong máy ảo dựa trên kỹ thuật mã mạng là hệ thống đảm bảo tối ưu truyền thông điệp từ nguồn một cách nhanh chóng, chính xác và đạt thông lượng cực đại tại tập đích. Ngoài ra, hệ thống cần phải đảm bảo tính gắn bó, nhất quán, trong xuốt đối với người sử dụng.*

Để đạt được những mục đích đặt ra, Luận án trình bày một số giải pháp và cũng là các đóng góp đã nghiên cứu:

- Giải pháp đảm bảo gắn bó dữ liệu trong hệ phân tán
- Giải pháp cung cấp tài nguyên truyền thông cho hệ phân tán trong máy ảo
- Giải pháp điều khiển tỷ lệ nguồn với mã mạng
- Giải pháp tối ưu truyền thông multicast với mã mạng

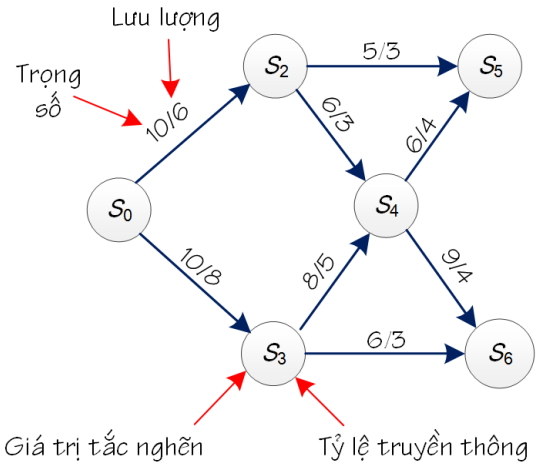
1.3.3. Giải pháp kỹ thuật

Việc mô tả các thành phần cơ bản trong Hình 1.23 đặt ra bài toán tổng quát quá trình cung cấp tài nguyên đám mây như sau: các thiết bị đầu cuối trong phần 5 đưa ra tập yêu cầu tài nguyên dùng chung $\mathbb{YC}$ thông qua môi trường truyền thông trong phần 4 đến bộ cung cấp tài nguyên trong phần 3. Bộ cung cấp tài nguyên trong phần 3 đảm nhiệm vai trò tiếp nhận, phân tích, tính toán, xử lý và hồi đáp tài nguyên đến các thiết bị truy cập với thời gian nhanh nhất trạng thái tài nguyên.



Hình 1.24: Mạng truyền thông phân tán ảo

Quá trình xử lý diễn ra bên trong hệ là tập các máy chủ ảo $\mathbb{U} = \{S_i | i = 0..n\}$ thông qua hệ thống truyền thông phân tán như Hình 1.24. Các máy chủ ảo S_i hoạt động hợp lực thông qua cơ chế truyền thông điệp để đáp ứng và phản hồi tài nguyên. Các thông điệp này cũng có thể được xét như là các tiến trình di chuyển trong hệ thống mạng. Việc thực thi các tiến trình phụ thuộc vào bộ cung cấp tài nguyên truyền thông để định tuyến từ nguồn đến đích.



Hình 1.25: Hệ thống mạng biểu diễn dưới dạng đồ thị

Hệ thống mạng ở Hình 1.24 có thể mô tả dưới dạng đồ thị $G = (\mathbb{U}, \mathbb{V})$ theo Hình 1.25, trong đó $\mathbb{U} = \{S_i | i = 0..n\}$ là tập các S_i và $\mathbb{V} = \{S_{ij} | i = 0..n, j = 1..m\}$ là

tập giữa các S_{ij} trong hệ, S_{ij} là liên kết giữa hai nút S_i và S_j liền kề trong hệ thống. Với mỗi S_{ij} có trọng số nguyên dương $ts_{S_{ij}} \geq 0$. Ký hiệu $\mathbb{P}$ là tập các phiên chuyển thông điệp trong hệ, đối với mỗi phiên $p \in \mathbb{P}$ là đường dẫn rời rạc từ nguồn $S_0 \in \mathbb{U}$ đến đích $S_{\mathbb{U}} \in \mathbb{U} \backslash \{S_0\}$. Đối với hệ thống đang xét, ký hiệu tập đích $\mathbb{D}$ thì $S_{\mathbb{U}} \in \mathbb{D}$. Ký hiệu lưu lượng ll^p của một phiên p truyền trong mạng là một giá trị nguyên dương, ta có $0 \leq ll^p_{S_{ij}} \leq ts_{S_{ij}}$.

Giải pháp cung cấp tài nguyên dùng chung cho hệ thống đám mây tập trung xử lý $\mathbb{YC}$ thông qua môi trường truyền thông dựa trên kỹ thuật mã mạng. Kỹ thuật này là một kỹ thuật kết nối mạng, trong đó truyền dữ liệu được mã hóa và giải mã để tăng thông lượng mạng, giảm độ trễ và tăng hiệu năng truyền thông. Trong mã mạng, các thuật toán đại số được áp dụng cho các gói dữ liệu để tích hợp và truyền trên nhiều kênh khác nhau, các gói tin nhận được giải mã tại tập đích. Quá trình thực thi yêu cầu băng thông ít hơn so với phương thức truyền trước đây nhưng đòi hỏi phải xử lý nhiều hơn tại các nút trung gian và nút đích.

Xuất phát từ nghiên cứu trong [140] của các tác giả Yeung và Zhang đưa ra khái niệm đầu tiên về mã mạng. Khái niệm mã mạng được trình bày như sau: *Trong một mạng chuyển mạch gói truyền thống, định nghĩa luồng dữ liệu trong là các "miếng" rời rạc từ nguồn đến đích. Tại đích hoặc các nút trung gian, các thông điệp gửi đi được chia thành các gói, mỗi trong số đó có chứa một số các thông điệp dữ liệu còn nguyên vẹn hoặc phân mảnh. Tất cả các gói dữ liệu không nhất thiết truyền theo định tuyến tương tự; nhưng tất cả chúng đến cùng một đích và ráp lại thành thông điệp ban đầu.*

Vấn đề chính của truyền thông là khi lưu lượng truy cập mạng tăng dễ diễn ra tắc nghẽn, dẫn đến độ trễ truyền lớn. Bên cạnh đó, các gói tin có xu hướng nhóm tại các nút xác định, đôi khi vượt quá khả năng xử lý chúng tại các nút; trong khi đó, các định tuyến và các nút khác có thể chưa được khai thác hết hiệu năng. Giải pháp chính của mã mạng là tối ưu định tuyến để đạt thông lượng cực đại và tránh trùng lặp thông tin từ nguồn đến tập đích đồng thời giải quyết các vấn đề tắc nghẽn, lạc gói tin, lỗi nút mạng,...

Trong mã mạng, bộ giải mã thay thế cho phần mềm định tuyến và chuyển mạch gói. Thay vì hướng các gói tin tới đích cuối của chúng, bộ giải mã truyền siêu dữ liệu (metadata) ở dạng chứng thực số về thông điệp theo nhiều đường dẫn cùng một lúc. Ngược lại, các siêu dữ liệu đến từ hai hoặc nhiều nguồn có thể được kết hợp thành một gói duy nhất. Giải pháp phân tán gói tin có thể làm tăng năng lực xử lý của một mạng bằng cách giảm số lượng nút truyền và mức độ tắc nghẽn. Khi bộ nhận có đủ chứng thực số, bộ giải mã có thể tính toán để tái thiết lập các thông điệp hoặc gói tin

truyền. Đối với một số các gói tin khi định tuyến bị mất hay bị cắt, thông điệp ban đầu được tái thiết lập nếu nhận đủ các chứng thực số.

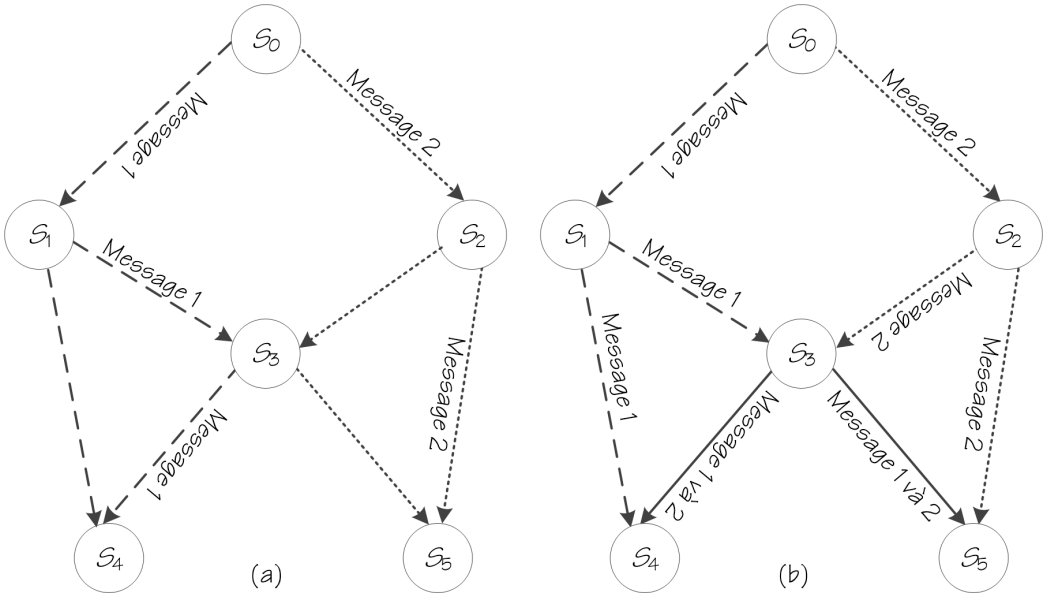
Trong kỹ thuật mã mạng, dữ liệu không những phụ thuộc vào thông điệp truyền mà còn đối với nội dung của các thông điệp khác diễn ra được chia sẻ định tuyến tại thời điểm truyền. Giải pháp của kỹ thuật mã mạng là trộn nhiều gói tin vào và định tuyến lại gói tin ra khi đi qua một nút mạng. Vì vậy, việc cải tiến thông lượng của kỹ thuật mã mạng phụ thuộc vào tô pô và khả năng sinh tắc nghẽn. Theo Hình 1.25, hai bài toán của kỹ thuật mã mạng là xử lý trên nút và xử lý trên cạnh. Bài toán xử lý trên cạnh với hai tham số chính là lưu lượng và trọng số. Bài toán xử lý trên nút là mã hóa và giải mã cùng với hai tham số là giá trị tắc nghẽn và tỷ lệ truyền thông.

Giải pháp kỹ thuật mã mạng Luận án phát triển đạt được tỷ lệ thông tin truyền tốt hơn so với tỷ lệ thông tin truyền thông trong hệ phân tán. Một số giải pháp của hệ phân tán hiện vẫn được triển khai theo kết nối ngang hàng (P2P) và truyền thông tin chủ yếu theo phương thức truyền unicast thể hiện ở Hình 1.26.a. Cách thức truyền này gửi từ nút nguồn đến nút đích theo định tuyến sẵn có. Mô tả trong Hình 1.26.a, nút nguồn S_0 truyền đến đích S_4 gói tin *Message* 1 theo định tuyến $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_4$, tương tự đối với S_5 có định tuyến là $S_0 \rightarrow S_2 \rightarrow S_5$. Cách thức định tuyến này dễ gây mất gói tin trong quá trình truyền nếu một điểm trong định tuyến bị sự cố và tỷ lệ gói tin nhận được thấp. Nếu nút đích S_4 hoặc S_5 muốn nhận được đầy đủ 2 gói tin thì chúng phải yêu cầu nút nguồn phát tiếp gói tin còn lại sau khi đã nhận gói tin trước.

Trường hợp mô hình mạng theo cách truyền multicast thể hiện trong Hình 1.26.b, nút đích S_4 nhận được gói tin theo hai tuyến khác nhau, tuyến thứ nhất $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_4$ và tuyến thứ hai $S_0 \rightarrow S_1 \rightarrow S_4$. Như vậy, gói tin tại nút đích có tỷ lệ nhận cao hơn so với cách truyền unicast. Nhược điểm của phương thức tuyền multicast là gói tin có thể bị trùng lặp tại tập đích dẫn đến lãng phí tài nguyên truyền thông. Giải pháp của Luận án sử dụng kỹ thuật mã mạng loại trừ gói tin trùng lặp bằng phép toán XOR trong quá trình định tuyến gói tin theo mô tả trong Hình 1.27.

Xét mạng mô tả trong Hình 1.27, chúng ta có các thông tin cơ bản: tập nút đích: S_4 và S_5; nút nguồn: S_0; các nút trung gian: S_1, S_2, S_3 và các liên kết: S_{01}, S_{02}, S_{13}, S_{14}, S_{23}, S_{25}, S_{34}, S_{35}; gói dữ liệu truyền: *Message 1* ($M1$), *Message 2* ($M2$) và $Message1 \oplus 2$ ($M_{1\oplus 2}$). Nút S_0 gửi đoạn mã thăm dò và các nút S_4 và S_5 nhận được chúng. Các nút trung gian có thể giám sát nội dung của các gói dữ liệu vào và gói để chuyển tiếp đến nút khác qua liên kết ra của chúng. Theo phép thử Bernoulli [17], mỗi liên kết bị mất một gói tin với xác suất chúng ta không biết. Luận án quan tâm đến dự đoán các xác suất mất trong tất cả các liên kết, ký hiệu xác suất $x_{01}, x_{02}, x_{13}, x_{14}, x_{23}, x_{25}, x_{34}, x_{35}$

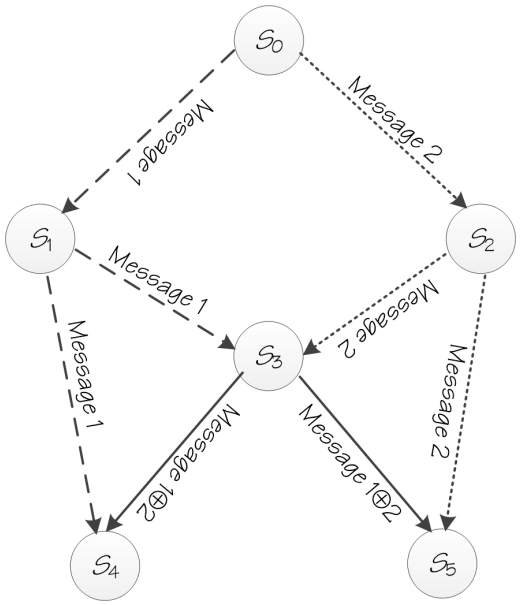
tương ứng với các liên kết trung gian.



Hình 1.26: Cách thức truyền unicast (a) và multicast (b)

Ý tưởng xử lý cơ bản của Luận án như sau: nút S_0 gửi đến nút S_1 một gói tin thăm dò có chứa các chuỗi nhị phân $M_1 = [10]$. Tương tự, nút S_0 gửi đến nút S_2 gói tin thăm dò $M_2 = [11]$. Nếu nút S_3 chỉ nhận được M_1 hoặc M_2, sau đó nút S_3 chỉ chuyển tiếp các gói tin nhận được đến nút đích S_4 hoặc S_5 thì tập các nút đích sẽ không nhận được đầy đủ dữ liệu. Nếu S_3 nhận được cả hai gói M_1 và M_2, sau đó nút S_3 tạo ra một gói tin mới kết hợp tuyến tính của chúng $M_{1\oplus 2} = [11]$ và chuyển tiếp đến tập nút đích, như vậy, nút đích S_4 và S_5 tiếp nhận đầy đủ các gói tin ban đầu. Tổng quan hơn, ta có $M_{1\oplus 2} = M_1 \oplus M_2$, trong đó sử dụng phép toán XOR các bit-wise, tất cả các hoạt động xảy ra cùng một khoảng thời gian.

Trong mỗi khoảng thời gian, mã thăm dò các gói dữ liệu (M_1, M_2) được gửi từ nút nguồn và có thể nhận được từ tập nút đích tùy thuộc vào thử nghiệm ngẫu nhiên: trên mỗi liên kết các nút trung gian, các gói dữ liệu truyền với xác suất x_{ij}. Các kết quả tại tập các nút đích S_4 và S_5 được trình bày trong hai cột bên trái của Bảng 1.12. Tám cột bên phải Bảng 1.12 hiển thị sự kết hợp của sự kiện mất (thể hiện giá trị 0) và thành công (thể hiện giá trị 1) trong các liên kết. Đối với các sự kiện diễn ra trên tập nút đích có 9 sự kiện được phái sinh theo gói tin thăm dò gửi từ nút đích. Theo Bảng 1.12 kết quả nhận được sự kiện thứ 8 đó là (M_1, M_2) là do các xác xuất ghi nhận được $x_{01} = 0, x_{02} = 1, x_{13} = 0, x_{14} = 0, x_{23} = 1, x_{25} = 1, x_{34} = 1, x_{35} = 1$. Tương tự, xác suất các trường hợp còn lại nằm trong 8 sự kiện quan sát được các xác suất liên kết mã thăm dò theo Bảng 1.12.



Hình 1.27: Cách thức truyền multicast kết hợp với mã mạng

Bảng 1.12: Mã thăm dò có thể quan sát được

TT	Tập nút đích		Tập nút trung gian							
	S_4	S_5	S_{01}	S_{02}	S_{13}	S_{14}	S_{23}	S_{25}	S_{34}	S_{35}
1	M_1	-	1	0	1	1	0	0	1	0
2	M_2	-	0	1	0	0	1	0	1	0
3	$(M_{1\oplus 2})$	-	1	1	1	1	1	0	1	0
4	-	M_1	1	0	1	0	0	0	0	1
5	-	M_2	0	1	0	0	1	1	0	1
6	-	$(M_{1\oplus 2})$	1	1	1	0	1	1	0	1
7	M_1	M_1	1	0	1	1	0	0	1	1
8	M_2	M_2	0	1	0	0	1	1	1	1
9	$(M_{1\oplus 2})$	$(M_{1\oplus 2})$	1	1	1	1	1	1	1	1

Nghiên cứu của Luận án là dự đoán xác suất từ các nội dung của các đầu dò nhận được tại các nút S_4 và S_5. Bằng cách lặp lại nhiều lần để thăm dò mỗi sự kiện xảy ra, sau đó sử dụng tiêu chuẩn dự đoán Maximum Likelihood (ML) trong [94] để suy ra tỷ lệ tổn thất liên kết cơ bản. Dự đoán ML xác định tỷ lệ mất liên kết có thể với xác suất cao hơn khi triển khai trên cây multicast. Kết quả đạt được tập dữ liệu cụ thể truyền multicast từ nguồn đến tập đích.

Cách tiếp cận sử dụng cây multicast xuất phát từ nguồn từ S_0 và đích tại S_4 và S_5 có những ưu điểm đó là phương pháp tiếp cận cây multicast sẽ giảm tối đa tỷ lệ mất gói tin các liên kết S_{01} và S_{13}, tương tự đối với S_{02} và S_{23}. Trong thử nghiệm, thuật toán gửi một mã thăm dò trên mỗi liên kết, đó là các yêu cầu tối thiểu để có thể bao trùm toàn bộ đồ thị. Bằng cách tiếp nhận không chỉ ở số lượng các đoạn mã thăm dò

nhận được mà còn ở nội dung thông tin, từ đó có thể suy luận thông tin bổ sung.

Hướng nghiên cứu của Luận án về *kỹ thuật mã mạng* tập trung vào các giải pháp:

- Giải pháp điều khiển tỷ lệ nguồn với mã mạng là một phần trong *bộ điều khiển cung cấp tài nguyên mã mạng* trong phần 3 của mô hình tổng quát nhằm phân chia gói tin đảm bảo tỷ lệ truyền từ nguồn đến tập đích.
- *Giải pháp tối ưu truyền thông phân tán dựa trên kỹ thuật mã mạng* cho phép tối ưu băng thông, tránh trùng lặp thông tin đồng thời đạt thông lượng cực đại tại tập đích.

Kết quả đánh giá của toàn bộ các giải pháp của Luận án dựa trên mô hình tổng quát trong Hình 1.23 thể hiện qua Bảng 1.13.

Tiểu kết Chương 1

Trên cơ sở tìm hiểu khái niệm, nghiên cứu chiến lược cung cấp tài nguyên, Luận án đã đưa ra mô hình tổng quát cho bài toán cung cấp tài nguyên trong hệ thống ảo hóa. Luận án đã đưa ra ví dụ hệ thống giám sát trực tuyến các phương tiện cơ giới đường bộ trình bày trong công bố số (1) và chương trình bãi đỗ xe trình bày trong công bố số (5). Dựa vào mô hình tổng quát trình bày trong công bố số (6), Luận án nêu lên một số giải pháp cơ bản nhằm cho phép tính toán, xử lý dữ liệu dùng chung trên đám mây. Nhóm giải pháp thứ nhất bao gồm cung cấp tài nguyên dùng chung cho hệ phân tán và cung cấp tài nguyên truyền thông phân tán. Nhóm giải pháp thứ nhất tập trung giải quyết vấn đề điều khiển cung cấp tài nguyên dùng chung trong hệ phân tán. Nhóm giải pháp thứ hai bao gồm điều khiển tỷ lệ nguồn với mã mạng và tối ưu truyền thông multicast dựa trên kỹ thuật mã mạng. Nhóm giải pháp thứ hai nhằm giải quyết bài toán tối ưu truyền thông cho hệ phân tán trong máy ảo dựa trên kỹ thuật mã mạng.

Bảng 1.13: Đánh giá tổng quát các nhóm giải pháp

Giảipháp	Hệ thống hiện có		Hệ thống nghiên cứu của luận án	
	Ưu điểm	Nhược điểm	Ưu điểm	Nhược điểm
Bộ cung cấp tài nguyên cho hệ phân tán	Sử dụng thuật toán 3PC trong giao dịch phân tán để đảm bảo tính gắn bó. Các giao dịch trao đổi thông điệp ít. Độ phức tạp thuật toán thấp.	Các giao dịch không sử dụng giá trị đồng hồ lô gic để so sánh, sắp xếp trật tự ưu tiên và xử lý đồng bộ tiến trình. Chi phí khôi phục giao dịch cao.	Thuật toán 4PCoDT đảm bảo tính gắn bó mạnh trong hệ phân tán. Thuật toán 4PCoDT sử dụng giá trị đồng hồ lô gic để xử lý đồng bộ hóa tiến trình và không có chi phí khôi phục giao dịch.	Trao đổi thông điệp nhiều dẫn đến tiêu tốn tài nguyên băng thông. Độ phức tạp thuật toán cao.
Bộ cung cấp tài nguyên truyền thông ảo hóa	Bộ cung cấp tài nguyên truyền thông ảo hóa cho phép mở rộng, tách các chức năng điều khiển và chuyển tiếp mạng, cho phép lập trình trực tiếp cho các ứng dụng và dịch vụ mạng.	Chương trình xây dựng cho bộ cung cấp tài nguyên truyền thông ảo hóa chưa hoàn chỉnh, vẫn còn thiếu tính năng, còn thiếu sự hỗ trợ. Các định tuyến chưa hoàn chỉnh, nhất là đối với phương thức truyền multicast, chưa xử lý hoàn chỉnh trùng lặp thông tin tại tập đích.	Bộ cung cấp tài nguyên truyền thông ảo hóa mở rộng dựa trên các chương trình sẵn có của hệ thống ảo hóa. Chương trình xây dựng bộ cung cấp tài nguyên truyền thông mở rộng từ NVF và SDN cho phép tối ưu các kênh truyền và xử lý trùng lặp tại tập đích.	Việc mở rộng chương trình xây dựng bộ cung cấp tài nguyên truyền thông tăng độ phức tạp xử lý tại các nút, bên cạnh đó bảo mật của gói tin truyền cần phải được quan tâm, xem xét.
Bộ điều khiển cung cấp tài nguyên mã mạng	Bộ điều khiển cung cấp tài nguyên mã mạng thuận lợi trong việc định tuyến, cải thiện thông lượng. Bộ điều khiển cung cấp tài nguyên mã mạng tối ưu nội dung truyền, tránh trùng lặp tại tập đích thông qua phép toán XOR.	Bộ điều khiển cung cấp tài nguyên mã mạng chỉ áp dụng cho phương thức truyền multicast, broadcast. Đối với các tô pô xây dựng chuẩn thì mã mạng mới phát huy hiệu năng đối với tối ưu thông lượng.	Bộ điều khiển cung cấp tài nguyên mã mạng áp dụng cho hệ phân tán triển khai trong hệ thống ảo hóa. Chương trình điều khiển cung cấp tài nguyên mã mạng mở rộng các thành phần trong mạng ảo (NVF và SDN) trên mô phỏng CloudSim. Để đạt được hiệu năng của mã mạng, Luận án đã trình bày hai thuật toán thêm và xóa liên kết nhằm rút gọn tô pô và thuật toán cải tiến Ford Fulkerson để tối ưu chi phí trong cây multicast. Bộ điều khiển cung cấp tài nguyên mã mạng tránh trùng lặp nội dung, tối ưu thông lượng tại tập đích.	Việc xây dựng bộ điều khiển cung cấp tài nguyên mã mạng thông qua mã và giải mã làm tăng độ phức tạp tại các nút trung gian và tập đích. Hạn chế của giải pháp triển khai làm chậm khả năng xử lý tại các nút trung gian và nút đích so với giải pháp truyền multicast.

Chương 2.

GIẢI PHÁP ĐIỀU KHIỂN CUNG CẤP TÀI NGUYÊN TRUYỀN THÔNG TRONG HỆ PHÂN TÁN

Hệ tập trung ra đời để giải quyết nhu cầu trao đổi thông tin người sử dụng trên toàn cầu. Tuy nhiên, hệ tập trung vẫn còn những nhược điểm cần phải giải quyết sau:

- Tắc nghẽn truyền thông khi số lượng truy cập vào hệ thống vượt nhu cầu cung cấp tài nguyên.
- Khả năng mở rộng tài nguyên phần cứng khi tài nguyên tới hạn hoặc không còn khả năng cung cấp.
- Cơ sở dữ liệu ở trạng thái cục bộ, khi xảy ra sự cố hệ thống không đảm bảo tính gắn bó đối với người sử dụng.
- Khi hệ thống gặp sự cố, hệ thống sẽ dừng hoạt động.

Hệ phân tán được hình thành trên cơ sở giải quyết các nhược điểm của hệ tập trung. Các nghiên cứu và triển khai hệ phân tán đã và đang được quan tâm nghiên cứu từ trước cho đến nay cho thấy hệ phân tán vẫn còn nhiều hướng mở để tiếp tục phát triển. Trên cơ sở đó, Luận án đã đi sâu vào các giải pháp của hệ phân tán như: giải pháp cung cấp tài nguyên truyền thông phân tán và giải pháp đảm bảo tính gắn bó. Các nhóm giải pháp giải quyết bài toán gắn bó dữ liệu trong hệ phân tán triển khai trên tầng SaaS trong mô hình tổng quát thể hiện ở Hình 1.23.

2.1. Giải pháp song song hóa thuật toán Lamport trong loại trừ tương hỗ phân tán

2.1.1. Song song hóa trong thuật toán Lamport

Mục 1.2.1.3 trình bày về vấn đề kỹ thuật trong nhãn thời gian lô gic dựa trên thuật toán Lamport. Các máy chủ hoạt động nhận và gửi thông điệp dựa trên đồng hồ cục bộ của mình theo cơ chế truyền unicast. Do đó, các máy chủ chỉ biết được trật tự từng phần trên máy chủ của mình và không nhận biết được các hoạt động trên máy chủ khác. Trật tự từng phần ảnh hưởng đến hoạt động tổng quát trong hệ phân

tán, hai vấn đề cơ bản bị tác động đó là: 1. Giá trị đồng hồ lô gic trên các máy chủ không nhất quán; 2. Tiến trình yêu cầu vào đoạn găng phải chờ đợi cho đến khi nhận đủ thông điệp có thể gây ảnh hưởng đến các máy chủ khác hoặc sai lệch khi tiến hành cập nhật dữ liệu.

Mục 1.2.1.3 cũng đã phân tích nhược điểm thuật toán Lamport bị động dựa trên trật tự từng phần để từ đó cải tiến thuật toán Lamport chủ động dựa trên trật tự tổng quát chặt chẽ. Để giải quyết bài toán trật tự từng phần, nghiên cứu của bài báo là song song hóa thuật toán Lamport để xây dựng trật tự tổng quát chặt chẽ trên các máy chủ theo Thuật toán 1.

Dựa trên truyền thông nhóm và thuật toán Lamport để đề xuất *song song hóa thuật toán Lamport* theo Thuật toán 1 của Luận án theo cách giải quyết sau: Mỗi máy chủ S đều có trang bị công tơ với các giá trị nguyên gọi là H_{S_i}, đồng hồ lô gíc tăng lên khi có sự kiện sk diễn ra trên một máy chủ bất kỳ. Một máy chủ i khi phát sinh sự kiện, máy chủ tăng giá trị lên 1 và gửi song song thông điệp yêu cầu được cung cấp giá trị đồng hồ đến tất cả các máy chủ còn lại theo thủ tục:

$$\text{GETLAMPORT}(S_i, H_{S_i}, act, sk)$$

tất cả các máy chủ sau khi nhận được thông điệp trên kiểm tra, so sánh với giá trị hiện tại $H_{S_{local}}$ của máy chủ mình $H_{S_i} = H_{S_{local}} + 1$ và phản hồi thông điệp chấp nhận giá trị với thủ tục:

$$\text{ACCEPTLAMPORT}(S_{local}, S_i, H_{S_i}, act, sk, boolean)$$

Sau khi tiếp nhận thông điệp phản hồi từ các máy chủ còn lại, nếu tham số thứ 5 nhận được đều có giá trị *true* thì tiến hành lấy $\max(H_{S_i})$, cập nhật lại giá trị đồng hồ theo $\max(H_{S_i})$. Giá trị mới được đánh dấu gắn cho sự kiện máy chủ yêu cầu và cập nhật lại đồng hồ. Thông điệp được thay đổi giá trị đồng hồ gửi song song đến các máy chủ theo thủ tục để xác nhận cho sự kiện sk theo thủ tục:

$$\text{UPDATELAMPORT}(S_i, H_{S_i}, act, sk)$$

Tất cả các máy chủ nhận so sánh và cập nhật lại giá trị đồng hồ theo giá trị đồng hồ hiện tại của máy chủ S_i cung cấp, đồng thời ghi nhận nhãn thời gian sự kiện sk cho máy chủ phát yêu cầu.

Kết quả của thuật toán là tất cả các máy chủ cùng thực hiện cập nhật giá trị đồng hồ và tuân thủ theo trật tự toàn phần như Hình 2.1. Kết quả song song hóa thuật

Thuật toán 1: *Song song hóa thuật toán Lamport*

Input : Máy chủ S_i, Giá trị đồng hồ lô gic H_{S_i}, hành động act, sự kiện sk

Output: Giá trị đồng hồ lô gic H_{S_i} đã cấp

Khởi tạo hoạt động $H_{S_{local}} = 0$, biến đếm $count$=0, $countCS[S_i, sk] = 0$, số lượng máy chủ S=N, lắng nghe sự kiện act;

Begin

if *(act=REQ-LAMPORT)* **then**

$H_{S_i} = H_{S_{local}} + 1$;

$act = REQ$;

Thiết lập thông điệp yêu cầu cung cấp giá trị đồng hồ lô gic: GetLamport($S_{local}, H_{S_{local}}, act, sk$);

return multicast(GetLamport(S_i, H_{S_i}, act, sk));

else if *(act=LAMPORT)* **then**

return H_{S_i};

procedure GetLamport(S_i, H_{S_i}, act, sk)

if *(act = REQ)* **then**

if *(*$H_{S_i} \leq H_{S_{local}}$*)* **then**

Xác định H_{S_i} bị sai, H_{S_i} đã được gán cho sự kiện khác;

act=REP;

return *multicast(AcceptLamport(*$S_{local}, S_i, H_{S_i}, act, sk, false$*))*;

else if *(*$H_{S_i} = H_{S_{local}} + 1$*)* **then**

Xác định H_{S_i} đúng;

act=REP;

return *multicast(AcceptLamport(*$S_{local}, S_i, H_{S_i}, act, sk, true$*))*;

end procedure

procedure AcceptLamport($S_{local}, S_i, H_{S_i}, act, sk, boolean$)

if *(act = REP)* **then**

if *($S_i = S_{local}$&&true)* **then**

$count = count + 1$;

if *()count = N - 1)* **then**

Xác nhận đủ số lượng thông điệp phản hồi *REP*;

$act = ACC$;

$count = 0$;

return *multicast(UpdateLamport($S_{local}, H_{S_{local}}, act, sk$))*;

else if *($S_i = S_{local}$&&false)* **then**

$H_{S_{local}} = \max(H_{S_i} + 1)$;

$act = REQ$;

Thiết lập lại thông điệp yêu cầu cung cấp với giá trị đồng hồ lô gic mới với sự kiện đang yêu cầu;

GetLamport($S_{local}, H_{S_{local}}, act, sk$);

end procedure

procedure UpdateLamport(S_i, H_{S_i}, act, sk)

if *(act = ACC)* **then**

$H_{S_{local}} = H_{S_i}$;

if *(sk = REP-CS)* **then**

$count[S_i, sk] = count[S_i, sk] + 1$;

if *($count[S_i, sk] = N - 1$)* **then**

CriticalSection(sk,S_i);

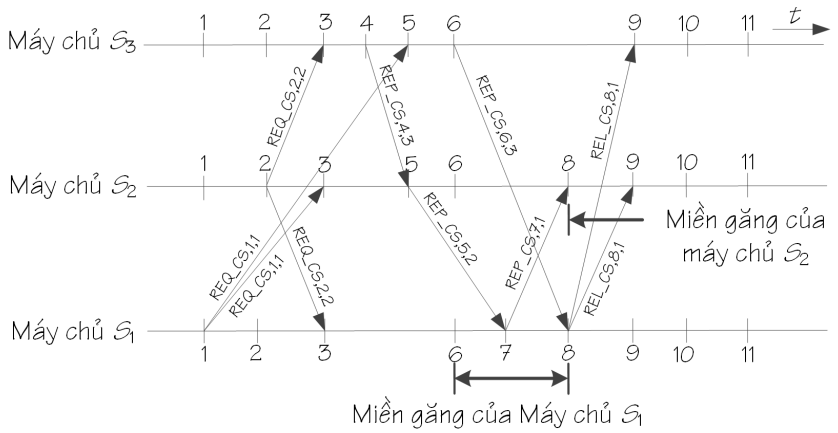
$count[S_i, sk] = 0$;

end procedure

toán Lamport của Luận án là điều kiện quan trọng đối với giải pháp loại trừ tương hỗ phân tán được trình bày trong Mục 2.1.2.

Sau khi thông điệp được gắn nhãn sẽ chuyển cho bộ cung cấp tài nguyên truyền thông xử lý. Bộ cung cấp tài nguyên truyền thông phân tán của Luận án tập trung vào các xử lý cơ bản là: cung cấp giá trị đồng hồ lô gíc, điều khiển lưu lượng và định tuyến truyền thông điệp qua các kênh truyền. Bộ cung cấp tài nguyên truyền thông cho phép các thuật toán được xây dựng và phát triển tham gia vào quá trình điều khiển nhằm tối ưu truyền thông điệp. Theo tác giả Li trong [80] trình bày cơ chế truyền thông điệp trong hệ phân tán thuộc bài toán NP-khó, do đó các giải pháp tối ưu chỉ mang giá trị đúng cho bài toán đang xét.

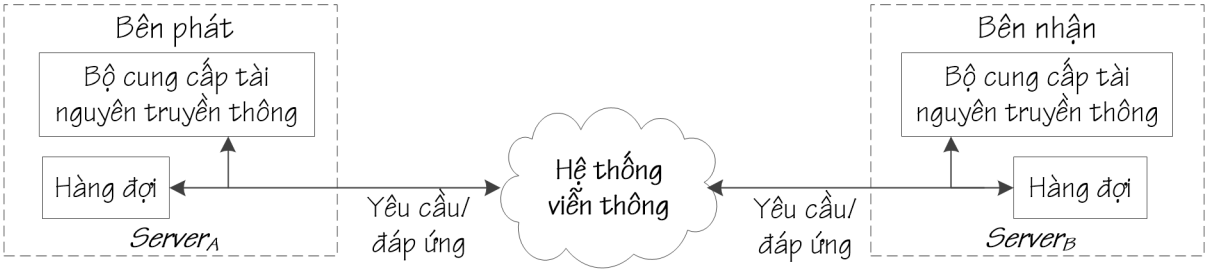
Truyền thông nhóm thực hiện truyền thông điệp dựa vào đường dẫn định tuyến đã biết để truyền từ nguồn đến đích. Truyền thông nhóm xét hai khía cạnh đó là *lưu lượng* và *các kênh truyền thông*. *Lưu lượng* thể hiện giá trị được cấp phát để truyền thông điệp, trong khi đó các *kênh truyền thông* được xem xét như là tài nguyên của bộ cung cấp tài nguyên truyền thông để truyền thông điệp. Đối với các máy chủ có nhiều kênh truyền thông thì hiệu quả truyền tốt hơn; tuy nhiên, vấn đề này còn phải phụ thuộc vào lưu lượng được cấp phát.



Hình 2.1: Trật tự tổng quát các thông điệp theo thuật toán Lamport sau khi cải tiến so với Hình 1.6

Bộ cung cấp tài nguyên truyền thông phân tán của Luận án dựa trên cơ sở truyền thông nhóm theo Hình 2.2. Tại mỗi máy chủ có hai thành phần: *hàng đợi* và *bộ cung cấp tài nguyên truyền thông*. *Hàng đợi* trong các máy chủ nhằm cho phép tiếp nhận các thông điệp đến là các tiến trình yêu cầu tài nguyên dùng chung. Các tiến trình yêu cầu tài nguyên được xử lý và sắp xếp theo giá trị đồng hồ; do đó, hệ phân tán có

thể giải quyết vấn đề tương tranh tài nguyên. *Bộ cung cấp tài nguyên truyền thông* trên mỗi máy chủ trong hệ đảm nhiệm vai trò vừa yêu cầu, vừa đáp ứng yêu cầu từ các Server khác trong hệ phân tán. Mục đích của truyền thông trong hệ phân tán là đảm bảo lưu lượng cực đại khi qua các nút để thông tin đến đích chính xác và nhanh chóng. Để giải quyết bài toán tối ưu truyền thông, Luận án *song song hóa thuật toán Ford Fulkerson trên đồ thị để tính toán lưu lượng cực đại trên các kênh truyền* trình bày tại Mục 3.3.2.3.



Hình 2.2: Cung cấp tài nguyên phân tán cho cặp yêu cầu/đáp ứng

2.1.2. Áp dụng song song hóa thuật toán Lamport để giải quyết loại trừ tương hỗ phân tán

Qua Hình 1.6, để các máy chủ nắm bắt được sự kiện diễn ra thì mỗi sự kiện được tạo truyền song song đến các máy chủ còn lại. Trên cơ sở song song hóa thuật toán Lamport trình bày trong Mục 2.1.1, thuật toán loại trừ tương hỗ phân tán thực hiện dựa trên cải tiến thuật toán Lamport được trình bày trong Thuật toán 2.

Bảng 2.1 thể hiện trật tự tổng quát trên tất cả các máy chủ, các sự kiện được các máy chủ khác nhận biết mặc dù không có sự kiện hoạt động trên máy chủ của chính mình. Theo Bảng 2.1, tại thời điểm giá trị đồng hồ lô gic 6, máy chủ S_1 nhận biết đã nhận đầy đủ 2 thông điệp *REP-CS*. Như vậy, đến thời thời điểm giá trị đồng hồ lô gic 6, máy chủ S_1 được phép vào miền găng. Tại thời điểm giá trị đồng hồ lô gic 8, máy chủ S_2 nhận biết được máy chủ S_1 đã xử lý xong và rời khỏi miền găng dựa trên thông điệp *REL-CS* nên tại thời điểm giá trị đồng hồ lô gic 8 máy chủ S_2 bắt đầu vào miền găng. Hình 2.1 thể hiện trật tự tổng quát các thông điệp theo thuật toán Lamport sau khi cải tiến. Sau khi áp dụng song song hóa thuật toán Lamport, tiến trình vào miền găng sơm hơn trong loại trừ tương hỗ.

Để đạt được kết quả trên, thuật toán Lamport cần được cải tiến sang phương thức truyền multicast. Nghiên cứu của Luận án đề ra giải pháp giải quyết trật tự tổng quát chặt chẽ trong hệ phân tán dựa trên song song hóa thuật toán Lamport trong loại trừ

tương hỗ phân tán. Các hoạt động của hệ phân tán phụ thuộc vào môi trường truyền thông có độ trễ là một trong những vấn đề quan tâm đó là trật tự thông điệp tiếp nhận và xử lý trên các máy chủ. Nghiên cứu của Luận án đảm bảo các trật tự thông điệp dựa trên thuật toán Lamport nhất quán giữa các máy chủ sao cho quá trình thực thi thông điệp dẫn đến gắn bó trong toàn hệ phân tán kết hợp thuật toán 4PCoDT trình bày trong Mục 2.2.

Bảng 2.1: Hoạt động diễn ra trên các máy chủ trong trật tự tổng quát chặt chẽ

Giá trị đồng hồ lô gic	**Máy chủ 1**	**Máy chủ 2**	**Máy chủ 3**
1	$S_1 \rightarrow S_2$:REQ-CS,1,1 $S_1 \rightarrow S_3$:REQ-CS,1,1	$S_1 \rightarrow S_2$:REQ-CS,1,1 $S_1 \rightarrow S_3$:REQ-CS,1,1	$S_1 \rightarrow S_2$:REQ-CS,1,1 $S_1 \rightarrow S_3$:REQ-CS,1,1
2	$S_2 \rightarrow S_1$:REQ-CS,2,2 $S_2 \rightarrow S_3$:REQ-CS,2,2	$S_2 \rightarrow S_1$:REQ-CS,2,2 $S_2 \rightarrow S_3$:REQ-CS,2,2	$S_2 \rightarrow S_1$:REQ-CS,2,2 $S_2 \rightarrow S_3$:REQ-CS,2,2
3	$S_2 \rightarrow S_1$:REQ-CS,2,2 $S_1 \rightarrow S_2$:REQ-CS,1,1 $S_2 \rightarrow S_3$:REQ-CS,2,2	$S_1 \rightarrow S_2$:REQ-CS,1,1 $S_2 \rightarrow S_3$:REQ-CS,2,2 $S_2 \rightarrow S_1$:REQ-CS,2,2	$S_2 \rightarrow S_3$:REQ-CS,2,2 $S_2 \rightarrow S_1$:REQ-CS,2,2 $S_1 \rightarrow S_2$:REQ-CS,1,1
4	$S_3 \rightarrow S_2$:REP-CS,4,3	$S_3 \rightarrow S_2$:REP-CS,4,3	$S_3 \rightarrow S_2$:REP-CS,4,3
5	$S_3 \rightarrow S_2$:REP-CS,4,3 $S_1 \rightarrow S_3$:REQ-CS,1,1	$S_3 \rightarrow S_2$:REP-CS,4,3 $S_1 \rightarrow S_3$:REQ-CS,1,1	$S_1 \rightarrow S_3$:REQ-CS,1,1 $S_3 \rightarrow S_2$:REP-CS,4,3
6	$S_3 \rightarrow S_1$:REP-CS,6,3	$S_3 \rightarrow S_1$:REP-CS,6,3 **Máy chủ S_1 vào miền găng**	$S_3 \rightarrow S_1$:REP-CS,6,3
7	$S_1 \rightarrow S_2$:REP-CS,7,1 **Máy chủ S_2 đã nhận đủ REP-CS, chờ lượt tiếp theo vào miền găng**	$S_1 \rightarrow S_2$:REP-CS,7,1	$S_1 \rightarrow S_2$:REP-CS,7,1
8	$S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1 $S_1 \rightarrow S_2$:REP-CS,7,1	$S_1 \rightarrow S_2$:REP-CS,7,1 $S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1 **Máy chủ S_2 vào miền găng**	$S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1 $S_1 \rightarrow S_2$:REP-CS,7,1
9	$S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1	$S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1	$S_1 \rightarrow S_2$:REL-CS,8,1 $S_1 \rightarrow S_3$:REL-CS,8,1

2.1.3. Hiệu năng thực thi song song hóa thuật toán Lamport

Dựa vào mô tả các tiến trình hoạt động trong miền găng và các tham số trình bày trong Mục 1.2.1.3 để đánh giá hiệu năng loại trừ tương hỗ phân tán theo Hình 2.3.

Tham số độ phức tạp thông điệp của song song hóa thuật toán Lamport được xác định dựa trên số lượng thông điệp yêu cầu trên một máy chủ cho mỗi thực thi miền

Thuật toán 2: *Cải tiến thuật toán loại trừ tương hỗ Lamport*

Input : Tiến trình
$tt(start,jeton,lamport_1,lamport_2,S_act,type,action,circle,content)$

Output: Trật tự tổng quát chặt chẽ tiến trình, tiến trình vào miền găng và loại trừ tương hỗ nhờ dấu

1 Khởi tạo hoạt động $H_{S_{local}} = 0$, biến đếm $countCS[S_i, sk] = 0$, $sk = \emptyset$, số lượng máy chủ S=N, hàng đợi req_queue, lắng nghe tiến trình tt;

2 **Begin**

$action = tt.action$;

if *(action=1)* **then**

$sk = REQ\text{-}CS$;

$act = REQ\text{-}LAMPORT$;

$H_{S_{local}}$=LAMPORT(S_local,act,sk);

$tt(start,jeton,H_{S_{local}},lamport_2,S_{local},type,action,circle,content)$;

$req_queue(tt)$;

multicast(RequestCriticalSection(sk,tt));

`RequestCriticalSection(`sk, tt`)`

$H_{S_{local}} = tt.lamport_1$;

$S_i = tt.S_act$;

if $(sk = REQ - CS \& H_{S_i} < H_{S_{local}})$ **then**

$sk = REP\text{-}CS$;

$act = REQ\text{-}LAMPORT$;

$H_{S_{local}}$=LAMPORT(S_local,act,sk);

$tt(start,jeton,H_{S_i},H_{S_{local}},S_i,type,action,circle,content)$;

else if $(sk = REP - CS \& H_{S_i} < H_{S_{local}})$ **then**

$sk = REP\text{-}CS$;

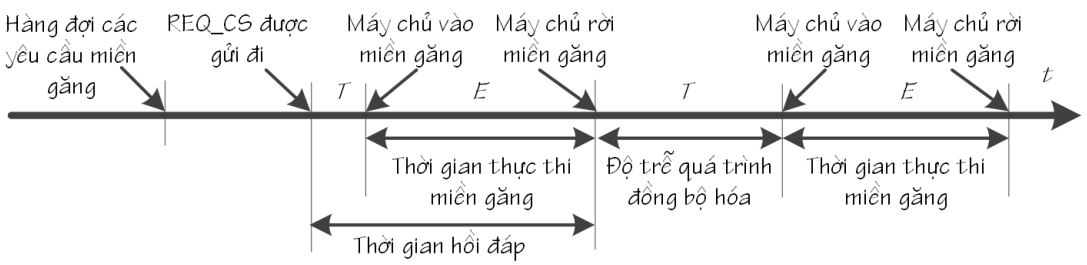
$act = REQ\text{-}LAMPORT$;

$H_{S_{local}}$=LAMPORT(S_local,act,sk);

$writelog(tt(start,jeton,H_{S_i},H_{S_{local}},S_{local},type,action,circle,content))$;

return RequestCriticalSection(sk,tt);

```
CriticalSection(sk, tt)
content = tt.content;
process(content);
remove_req_queue(tt);
sk = REL-CS;
act = REQ-LAMPORT;
H_Slocal=LAMPORT(Slocal,act,sk);
H_Slocal=LAMPORT(Slocal,act,sk);
tt(start,jeton,lamport1,H_Slocal,Si,type,action,circle,content);
return multicast(NextCriticalSection());
NextCriticalSection()
if (req_queue != empty) then
    pop_req_queue(tt);
    return CriticalSection(sk,tt)
```



Hình 2.3: Mô tả các tiến trình hoạt động trong miền găng

găng. Đối với song song hóa thuật toán Lamport yêu cầu (N - 1) thông điệp REQ, (N - 1) thông điệp REP, (N - 1) thông điệp ACC, do đó, thuật toán yêu cầu 3(N - 1) thông điệp. Đối với thuật toán cải tiến loại trừ tương hỗ của bài báo yêu cầu (N - 1) thông điệp REQ-CS và không xét thông điệp REP-CS và REL-CS trong quá trình nhận, do đó, thuật toán yêu cầu (N - 1) thông điệp khi vào miền găng. Nguyên nhân độ phức tạp thông điệp thuật toán loại trừ tương hỗ thấp bởi vì khi áp dụng song song hóa thuật toán Lamport, các thông điệp REP-CS và REL-CS đã được nhận biết và đánh dấu khi yêu cầu giá trị đồng hồ lô gic trên các máy chủ. Do đó, máy chủ yêu cầu vào miền găng không cần phải đợi tiếp nhận đủ thông điệp REP-CS và REL-CS. Ngoài ra, trong quá trình truyền thông điệp trong hệ thống, thông điệp REP-CS và REL-CS bị phân mảnh hoặc thất lạc không ảnh hưởng đến quá trình vào miền găng của máy chủ.

Tham số độ trễ quá trình đồng bộ hóa ký hiệu là T được xác định dựa trên khoảng thời gian yêu cầu sau khi máy chủ bắt đầu phát thông điệp yêu cầu REQ-CS cho đến khi phát thông điệp rời khỏi miền găng REL-CS hoặc máy chủ rời miền găng để nhường cho máy chủ tiếp theo vào miền găng. Trong trường hợp tải cao, máy chủ tiếp theo được thực hiện miền găng tức thì sau khi máy chủ trước vừa rời khỏi miền găng. Tức có nghĩa là các máy chủ yêu cầu vào miền găng đã nhận đủ thông điệp phản hồi theo song song hóa thuật toán Lamport trong Thuật toán 1. Trong điều kiện lý tưởng, $T = 0$.

Tham số thời gian hồi đápký hiệu là H được xác định là khoảng thời gian từ khi gửi yêu cầu vào miền găng cho đến khi ra khỏi miền găng. Theo mô tả trong Hình 2.3, thời gian hồi đáp được tính bằng Công thức (2.1) theo [62]:

$$H = T + E \tag{2.1}$$

trong đó: T là độ trễ quá trình đồng bộ hóa và E là thời gian thực thi miền găng. Đối với trường hợp áp dụng song song hóa thuật toán Lamport, máy chủ được phép vào miền găng ngay tại thời điểm máy chủ cuối cùng bắt đầu phản hồi thông điệp REP-CS sau thông điệp REQ-CS. Như vậy, thời gian hồi đáp được xác định:

1. Đối với trường hợp chờ tiếp nhận máy chủ cuối cùng bắt đầu phát thông điệp REP-CS, nghĩa là T = (N - 1). H = (N - 1) + E.
2. Đối với trường hợp đã tiếp nhận đủ thông điệp REP-CS và đang nằm trong hàng đợi chờ vào miền găng trong lượt tiếp theo, nghĩa là $T = 0$. $H = E$.

Tham số thông lượng hệ thống ký hiệu là A được xác định dựa trên tỷ lệ mà hệ

thống thực thi các yêu cầu trong miền găng được tính bằng Công thức (2.2) theo [62]:

$$A = \frac{1}{H} \tag{2.2}$$

trong đó: H là thời gian hồi đáp. Đối với trường hợp $T = (N - 1)$ thì $A = 1/(N - 1) + E$. Đối với trường hợp $T = 0$ thì $A = 1/E$.

Đối với việc áp dụng song song hóa thuật toán Lamport trong thực thi miền găng, thời gian hồi đáp của tiến trình yêu cầu miền găng và độ trễ quá trình đồng bộ hóa được rút ngắn. Kết quả so sánh các thuật toán thể hiện trong Bảng 2.2, thuật toán Lamport cải tiến khi áp dụng song song hóa thuật toán Lamport đạt được hiệu năng loại trừ tương hỗ phân tán cao so với thuật toán Lamport và Ricart-Agrawala.

Bảng 2.2: So sánh hiệu năng của thuật toán Lamport cải tiến trong loại trừ tương hỗ phân tán

Thuật toán	**Thời gian hồi đáp**	**Độ trễ đồng bộ**	**Thông điệp tải thấp**	**Thông điệp tải cao**
Lamport	$2T + E$	T	$3(N - 1)$	$3(N - 1)$
Ricart-Agrawala	$2T + E$	T	$2(N - 1)$	$2(N - 1)$
Lamport cải tiến	$T + E$	T	$(N - 1)$	$(N - 1)$

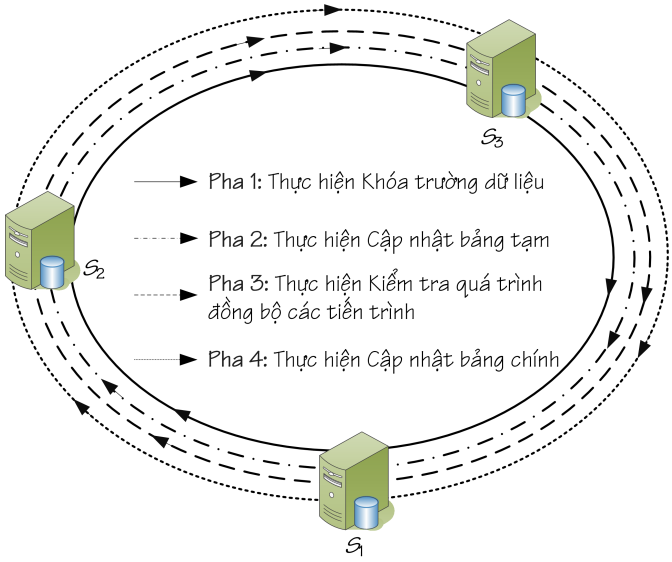
Song song hóa thuật toán Lamport cho phép thiết lập một trật tự tổng quát chặt chẽ và ghi dấu các sự kiện diễn ra trên các máy chủ. Thuật toán cải tiến gán dấu cho sự kiện yêu cầu 3(N - 1) thông điệp. Khi áp dụng song song hóa thuật toán Lamport trong thuật toán loại trừ tương hỗ, tiến trình đi vào miền găng yêu cầu (N - 1) thông điệp. Do đó, giải pháp cải tiến của Luận án đạt hiệu năng cao trong cải tiến thuật toán loại trừ tương hỗ phân tán. Giải pháp này là một phần trong thuật toán đảm bảo tính gắn bó trong thuật toán 4PCoDT.

2.2. Đề xuất thuật toán 4PCoDT điều khiển cung cấp tài nguyên trong hệ phân tán triển khai trong máy ảo

Giải pháp điều khiển cung cấp tài nguyên dùng chung trong hệ phân tán triển khai trong máy ảo dựa vào phát triển các hạn chế trong Mục 1.2.1.4 của Luận án. Ngoài ra, giải pháp được kế thừa và phát triển dựa trên thuật toán dành cho nhóm kín, dựa trên quyền và di chuyển tuần tự đảm bảo trật tự toàn phần trong hệ phân tán trình bày trong Mục 1.2.1.2. Hệ phân tán thực hiện đảm bảo gắn bó thực hiện

trên vòng tròn ảo thể hiện qua Hình 2.4. Các thông điệp di chuyển qua các máy chủ $S_i|i = 1..n$ thành vòng tròn gọi là vòng tròn ảo được trích dẫn trong [65], mỗi vòng thực hiện chức năng gọi là một pha trong tiến trình hoạt động.

Tính gắn bó trong hệ phân tán phải đạt được hai yếu tố cơ bản: 1. dữ liệu trên các máy chủ phải nhất quán và 2. các máy chủ thực hiện cùng một giải thuật. Các thông điệp truy cập đến hệ phân tán phải qua một giải thuật cơ bản, đó là giao dịch thể hiện qua thuật toán 4PCoDT mô tả trình tự hoạt động trên từng máy chủ. Thuật toán 4PCoDT được thực hiện trong hệ điều hành phân tán.



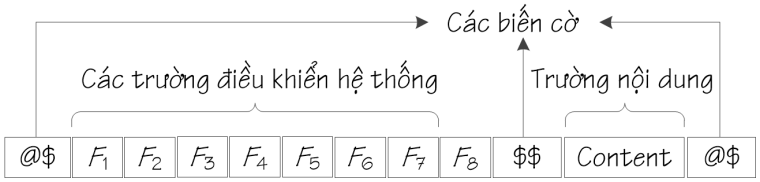
Hình 2.4: Thông điệp di chuyển theo vòng tròn ảo

Sự hợp lực, giám sát và điều khiển trong hệ phân tán chủ yếu dựa vào cơ chế truyền thông điệp. Do đó, việc xây dựng một cấu trúc thông điệp riêng biệt cho hệ là điều cần thiết. Thông qua cấu trúc thông điệp, hệ phân tán có thể phân biệt thông điệp dành cho ứng dụng đang thực hiện, đồng thời tách các trường điều khiển và nội dung thông điệp. Dựa vào cấu trúc thông điệp có thể nghiên cứu các giải pháp tăng tính bảo mật cho hệ phân tán dựa vào cấu trúc riêng và mã hóa thông điệp khi truyền trong hệ thống mạng.

Các trường điều khiển được tách trong thông điệp sẽ được cập nhật lại và lưu lại vết khi di chuyển trong hệ phân tán. Các quá trình hoạt động và đồng bộ hóa trong hệ phân tán dựa vào các trường trong thông điệp để đảm bảo tính gắn bó trong toàn hệ.

Bảng 2.3: Nội dung các trường điều khiển trong thông điệp

Trường	Ý nghĩa
F_1	Thông tin Server bắt đầu vòng tròn ảo [65], $F_1 = \overline{1..n \times 4}$ (n là số lượng máy chủ).
F_2	Giá trị Jeton [65] hiển thị cập nhật thông tin các Server trong hệ thống, độ dài của giá trị chính là n. Mỗi máy chủ sẽ chiếm giữ giá trị tại vị trí của máy chủ mình theo quy định. $F_2[i] = 0 \rightarrow 4, i = \overline{1..n}$, nếu giá trị không thay đổi thì xem như máy chủ đó bị sự cố và rời khỏi vòng tròn ảo.
F_3	Giá trị đồng hồ lô gíc [71] được gán và thiết lập chế độ ưu tiên thông điệp.
F_4	Giá trị đồng hồ lô gíc hiện tại được gán. Nếu $F_3 = F_4$ thì đây là thông điệp bắt đầu phát yêu cầu tài nguyên, ngược lại đó là thông điệp di chuyển trong hệ thống.
F_5	Tên máy chủ được cập nhật khi di chuyển trong hệ thống.
F_6	Nội dung tương ứng với pha giao dịch trong hệ thống, giao dịch thực hiện 4 pha cơ bản để đảm bảo tính nhất quán dữ liệu.
F_7	Các trạng thái hành động tương ứng với pha giao dịch trong hệ thống $F_7 = 1 \rightarrow 4$, kết thúc của 1 vòng di chuyển, F_7 tăng lên một đơn vị.
F_8	Giá trị vòng thông tin đăng ký, quá trình diễn ra đăng ký hoàn tất của một giao dịch giá trị sẽ tăng lên một. Trong quá trình hoạt động xảy ra lỗi đối với một máy chủ, khi khôi phục và vào lại vòng tròn, máy chủ đó chỉ cập nhật lại dữ liệu từ vòng bị lỗi cho đến vòng hiện tại dựa vào giá trị của trường Content.



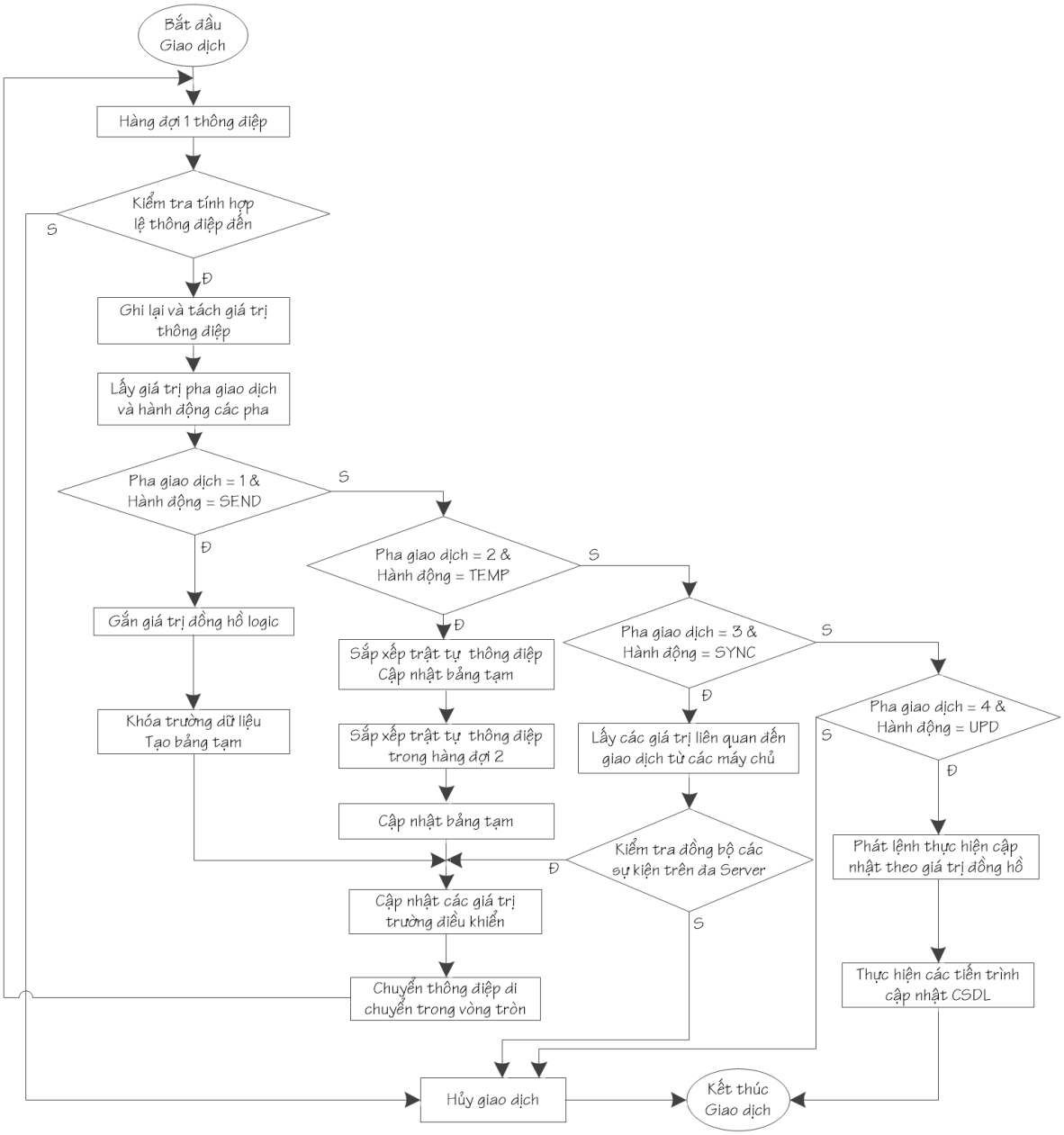
Hình 2.5: Cấu trúc thông điệp di chuyển theo vòng tròn ảo của hệ phân tán

Sau khi xây dựng thông điệp cho hệ thống, hệ phân tán điều khiển truyền thông điệp qua môi trường truyền thông đảm bảo các gói tin truyền đến tập đích nhanh chóng và chính xác, đảm bảo thời gian thực.

Cấu trúc của thông điệp dựa trên hai thành phần chính: các *trường điều khiển hệ thống* và *trường nội dung* tác động đến tài nguyên được tách lập bởi *các biến cờ* thể hiện qua Hình 2.5.

Các trường điều khiển được tách thành 8 trường, nội dung các trường mô tả trong Bảng 2.3. Các giá trị từ F_1 đến F_8 được cập nhật khi di chuyển qua mỗi máy chủ bên trong hệ. Trong đó, giá trị trường F_3 là trường quan trọng để so sánh giá trị ưu tiên nhằm giải quyết tương tranh tài nguyên và đồng bộ hóa các tiến trình trong hệ phân tán.

Dựa vào cấu trúc thông điệp, các quá trình điều khiển và giám sát thực hiện dựa trên các trường điều khiển hệ thống. Khi di chuyển qua các máy chủ, thông điệp được lưu vết và so khớp với các máy chủ khác đồng thời kiểm tra quá trình nhất quán dữ liệu sau khi khắc phục sự cố và tham gia vào hoạt động trong hệ.



Hình 2.6: Thuật toán 4PCoDT đảm bảo gắn bó trong hệ phân tán

Để đảm bảo tính gắn bó, dựa trên thuật toán 3PC [99] Luận án đề xuất thuật toán xử lý các pha giao dịch thể hiện ở Hình 2.6. Trong *pha 1*, dựa vào song song hóa thuật toán Lamport trình bày trong Mục 2.1.1 của Luận án để lấy giá trị đồng hồ lô gíc và gán cho thông điệp tại trường F_3 và F_4. Đối với thông điệp yêu cầu tài nguyên

thì giá trị F_3 chính là giá trị khởi đầu cho quá trình đăng ký tài nguyên - giá trị này không thay đổi khi di chuyển trong hệ thống - đồng thời dùng để so sánh khi có nhiều tiến trình cùng tương tranh một tài nguyên. Kết thúc pha 1, trường dữ liệu tại tài nguyên đăng ký bị khóa và một bảng dữ liệu tạm được tạo ra nhằm để ghi lại toàn bộ quá trình đăng ký trước khi cập nhật bảng chính. Khi trường dữ liệu bị khóa, tài nguyên được cấp phát duy nhất cho tiến trình đang chiếm giữ; khóa mở khi và chỉ khi tài nguyên bị quá hạn xử lý hoặc tiến trình có giá trị ưu tiên cao hơn thông qua giá trị đồng hồ lô gíc.

Trong *pha 2* sử dụng hai hàng đợi q_1 và q_2, q_1 tiếp nhận các thông điệp đến. F_3 đảm nhiệm vai trò là giá trị được sắp xếp trong q_2 tại pha 2. Kết quả đầu ra của q_2 là các thông điệp đã được sắp xếp theo trật tự giá trị F_3. Trong quá trình sắp xếp, nếu giá trị nào bị thiếu thì máy chủ yêu cầu phát lại thông điệp đó từ máy chủ còn lại trong hệ thống. Kết thúc pha 2, bảng dữ liệu tạm được cập nhật trên tất cả các máy chủ; nếu pha này bị hủy, chỉ cần xóa bảng tạm và mở khóa trường dữ liệu để giải phóng tài nguyên.

Pha 3 là pha nhằm kiểm tra quá trình đăng ký đảm bảo tài nguyên được cấp phát duy nhất, pha này so khớp toàn bộ quá trình hoạt động của giao dịch đang xử lý tại tất cả các máy chủ. Nếu quá trình so khớp không đúng, lập tức hủy giao dịch.

Pha 4 thực hiện xử lý tuần tự các tiến trình sau khi sắp xếp trong pha 2. q_2 đảm bảo giá trị đồng hồ lô gíc của thông điệp tại đầu ra của hàng đợi nhất thiết phải giống nhau. Lệnh yêu cầu cập nhật đến tất cả máy được thực hiện bởi máy chủ đang chiếm giữ giá trị F_3, đồng thời tăng giá trị F_3 lên một đơn vị.

Kết thúc quá trình thực hiện qua thuật toán 4PCoDT, dữ liệu trên các bảng đảm bảo tính nhất quán và phòng tránh tương tranh dữ liệu.

2.3. Triển khai giải pháp gắn bó trong hệ phân tán

Ban đầu phần mềm DSSim không có mạng lớp phủ. Mạng bao phủ là một mạng máy tính được xây dựng trên nhiều mạng khác. Do đó, mạng bao phủ sẽ cung cấp cái nhìn trừu tượng về toàn bộ mạng. Vì vậy, việc cải tiến và mở rộng DSSim có thể mô phỏng hàng ngàn nút thể hiện ở mức vật lý sẽ giúp theo dõi hoạt động toàn bộ hệ thống. Các nút trong mạng bao phủ có thể được thiết kế như một tập hợp các nút trung gian một-một tương ứng với tập hợp các nút vật lý được chọn. Các nút trong mạng bao phủ có thể được coi là kết nối bằng các liên kết ảo hoặc lô gíc; mỗi liên kết tương ứng với một đường dẫn, có thể thông qua nhiều liên kết vật lý trong lớp mạng bên dưới.

```
run:
Khai pha to po - Mo phong...
========================================

Dang tao to po...
Tong so nut: 100
Tong so lien ket: 197
token = 'Token[n=5.0], line 3'
So AS: 5
So nut tren AS: 20
Dang doc nut...
ID Nut = 0 toa do x = 536 toa do y = 233 nut AS = -1
ID Nut = 1 toa do x = 346 toa do y = 720 nut AS = -1
ID Nut = 2 toa do x = 534 toa do y = 624 nut AS = -1
...
ID Nut = 98 toa do x = 893 toa do y = 229 nut AS = -1
ID Nut = 99 toa do x = 661 toa do y = 284 nut AS = -1
Hoan tat ..... Dang doc cung....
token = 'Token[n=0.0], line 108'
ID lien ket = 0 tu nut = 0 den nut = 1 delay = 522.7513749384118 bang thong  = 10.0
ID lien ket = 1 tu nut = 0 den nut = 2 delay = 391.0051150560565 bang thong  = 10.0
ID lien ket = 2 tu nut = 1 den nut = 2 delay = 211.09239683134015 bang thong  = 10.0
...
ID lien ket = 195 tu nut = 99 den nut = 33 delay = 750.3392566033048 bang thong  = 10.0
ID lien ket = 196 tu nut = 99 den nut = 88 delay = 217.86234185833956 bang thong  = 10.0
Them nut logic...
Dang tao nut logic so 0 ...
Dang tao nut logic so 1 ...
Dang tao nut logic so 2 ...
Dang tao nut logic so 3 ...
Dang tao nut logic so 4 ...
Chuyen doi nut và lien ket den do thi...
```

Hình 2.7: DSSim đọc thông số tô pô

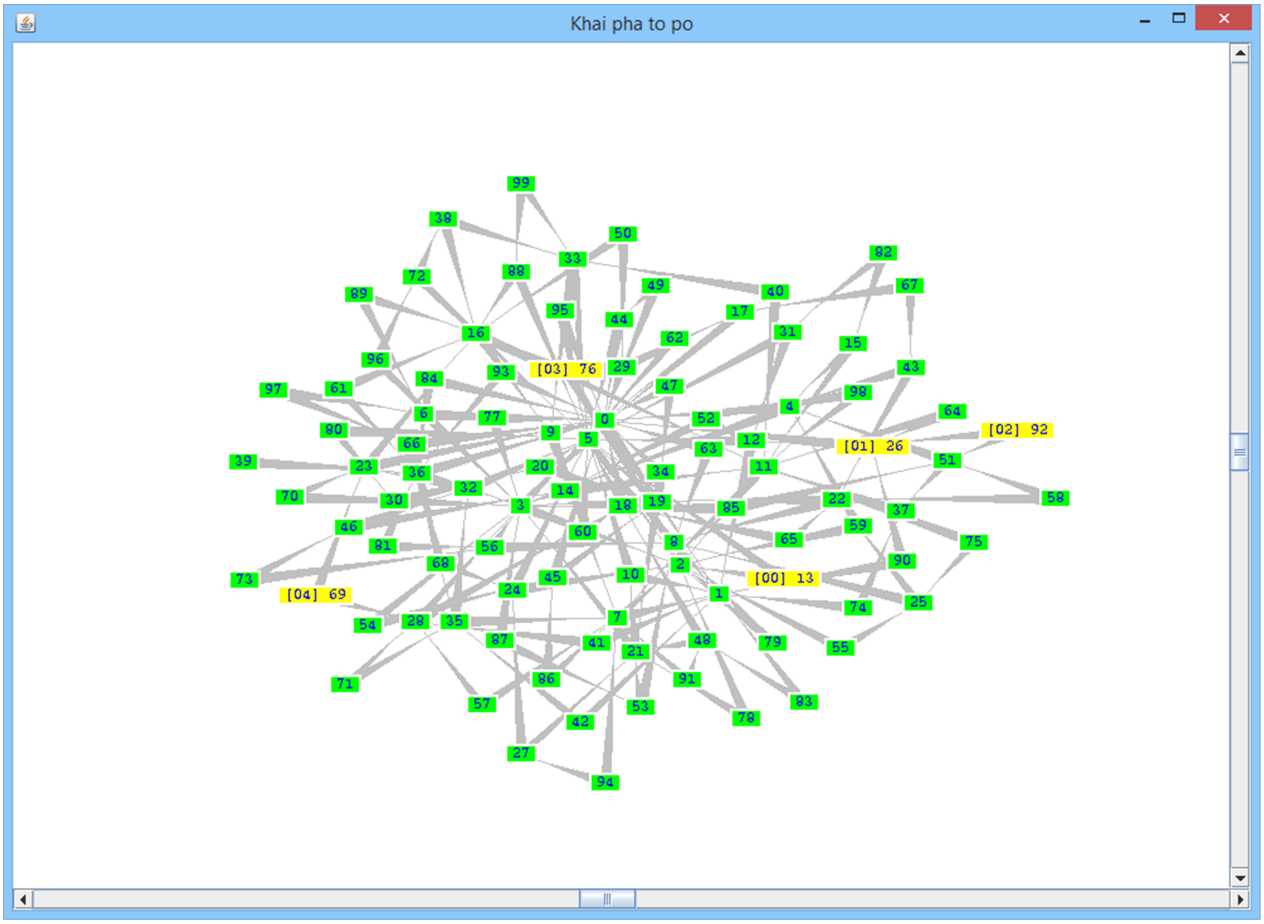
```
Nut dang chay = [01]
  dia chi nut = [01]
Thong diep da nhan = [02]->[01], "[02]"
nut: [01]
nut: [00]
Thong diep da gui = [01]->[02], "[01]"
thong tin su kien = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[01]",node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 472, data = [01]->[00], "[01]",node logic = null]
thong tin su kien = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[01]",node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 406, data = [04]->[03], "[04]",node logic = null]
Thong diep da gui = [01]->[02], "[00]"
thong tin su kien = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 472, data = [01]->[00], "[01]",node logic = null]
thong tin su kien = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 406, data = [04]->[03], "[04]",node logic = null]
thong tin su kien = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien nhan = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[01]",node logic = [01]]
nut: [01]
thong tin su kien = [su kien = MONITORING_LLINK_CREATION, nhan thoi gian = 246, data =
[Ljava.lang.Object;@63c12fb0,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 472, data = [01]->[00], "[01]",node logic = null]
thong tin su kien = [su kien = MONITORING_LLINK_CREATION, nhan thoi gian = 246, data =
[Ljava.lang.Object;@63c12fb0,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien = [su kien = MONITORING_LLINK_CREATION, nhan thoi gian = 246, data =
[Ljava.lang.Object;@63c12fb0,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 406, data = [04]->[03], "[04]",node logic = null]
thong tin su kien = [su kien = MONITORING_PROP_CHANGED, nhan thoi gian = 246, data = {knownHosts=[[01], [00],
[02]], myAddress=[01]},node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 472, data = [01]->[00], "[01]",node logic = null]
thong tin su kien = [su kien = MONITORING_PROP_CHANGED, nhan thoi gian = 246, data = {knownHosts=[[01], [00],
[02]], myAddress=[01]},node logic = [01]]
thong tin su kien nhan = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien = [su kien = MONITORING_PROP_CHANGED, nhan thoi gian = 246, data = {knownHosts=[[01], [00],
[02]], myAddress=[01]},node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 406, data = [04]->[03], "[04]",node logic = null]
thong tin su kien = [su kien = MONITORING_PROP_CHANGED, nhan thoi gian = 246, data = {knownHosts=[[01], [00],
[02]], myAddress=[01]},node logic = [01]]
thong tin su kien nhan = [su kien = MONITORING_LLINK_CREATION, nhan thoi gian = 246, data =
[Ljava.lang.Object;@63c12fb0,node logic = [01]]
thong tin su kien = [su kien = NODE_END, nhan thoi gian = 246, data = null,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 472, data = [01]->[00], "[01]",node logic = null]
thong tin su kien = [su kien = NODE_END, nhan thoi gian = 246, data = null,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_NODE_SEND, nhan thoi gian = 246, data = [01]->[02], "[00]",node logic = [01]]
thong tin su kien = [su kien = NODE_END, nhan thoi gian = 246, data = null,node logic = [01]]
thong tin su kien nhan = [su kien = MONITORING_PROP_CHANGED, nhan thoi gian = 246, data = {knownHosts=[[01],
[00], [02]], myAddress=[01]},node logic = [01]]
thong tin su kien = [su kien = NODE_END, nhan thoi gian = 246, data = null,node logic = [01]]
thong tin su kien nhan = [su kien = MSG_ROUTE_RECV, nhan thoi gian = 406, data = [04]->[03], "[04]",node logic = null]
Thoi gian hien tai = 245
```

Hình 2.8: Các sự kiện trong chương trình mô phỏng DSSim

2.3.1. Các hoạt động hệ phân tán

Trong các hệ thống mạng lớn thường xuyên xảy ra lỗi trong quá trình hoạt động. Mô phỏng lỗi là một trong những yếu tố quan trọng không thể thiếu trong mô phỏng thực tế. Vì vậy, ta cần phải mô phỏng lỗi ngẫu nhiên phát sinh của các hệ thống mạng. Các thuật toán trong các hệ thống phân tán cần có để đối phó với các lỗi mạng. Vì vậy, mô phỏng của lỗi là rất quan trọng, mô phỏng có thể tự động phục hồi từ lỗi. Phục hồi lỗi đạt được bằng cách sửa đổi các bảng định tuyến, các gói tin sẽ được định

tuyến theo bảng định tuyến mới.



Hình 2.9: Giao diện thể hiện kết quả thực thi tô pô trên công cụ mô phỏng DSSim

Các lớp chính trong quá trình thực thi DSSim là: SimModel, SimEvent, DSSim và Tô pô. Lớp SimModel định nghĩa các thuộc tính của mạng được mô phỏng bao gồm bốn thuộc tính là: mô hình định tuyến, ánh xạ nút, xử lý độ trễ nút và xử lý độ trễ định tuyến. Mô hình định tuyến xác định chiến lược được sử dụng để tính các bảng định tuyến trong lớp vật lý. Ánh xạ nút chứa thông tin về tất cả các nút lô gíc trong mạng ánh xạ các nút lô gíc tương ứng nút vật lý. Xử lý độ trễ nút là thời gian cần thiết để xử lý thông điệp tại nút đó. Xử lý độ trễ định tuyến là tổng thời gian cần thiết xử lý thông điệp của các nút trung gian để định tuyến thông điệp tới nút đích. DSSim mô phỏng dựa trên sự kiện rời rạc, vì vậy mọi hành động diễn ra trong hệ được xử lý như một sự kiện. Lớp SimEvent bao gồm các tập các sự kiện được xác định trước trong DSSim. DSSim cung cấp một số tô pô được xác định trước như tô pô vòng tròn ảo, tô pô ngẫu nhiên,...

DSSim phân biệt giữa tô pô mạng vật lý được tạo ra bởi bộ tạo tô pô và tô pô mạng lô gíc đạt được sau khi ánh xạ các nút tầng ứng dụng thực thi thuật toán phân

tán trên các nút vật lý. Các nút lô gíc truyền thông qua cơ chế truyền thông điệp, mô phỏng DSSim định tuyến các thông điệp trong tô pô mạng vật lý. DSSim sử dụng thuật toán phân cấp định tuyến véc tơ khoảng cách hai lớp tương tự định tuyến Internet trong các hệ thống tự trị.

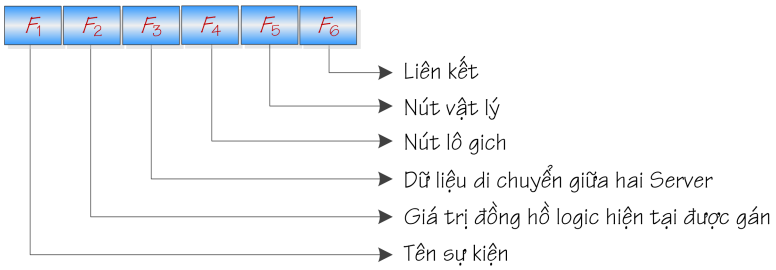
Để thực hiện mô phỏng các bước thực hiện theo quy trình:

- *Bước 1*: khai báo số nút ảo thực hiện việc tiếp nhận và xử lý thông điệp.
- *Bước 2*: khai báo đường dẫn đến tập tin đã được xuất ra từ chương trình BRITE.
- *Bước 3*: thực hiện chạy chương trình mô phỏng.
- *Bước 4*: nhận kết quả thực hiện qua 2 dạng: đồ họa hiển thị các nút hoạt động trên mặt phẳng theo Hình 2.9, các hoạt động và sự kiện diễn ra bên trong hệ theo Hình 2.7 và 2.8.

Bảng 2.4: Các sự kiện đối với nút trong mô phỏng phân tán

STT	**Tên sự kiện**	**Ý nghĩa**
1	UNKNOWN	Không xác định sự kiện
2	ALL	Tất cả sự kiện
3	MSG_NODE_SEND/MSG_NODE_RECV	Nút gửi/nhận thông điệp
4	MSG_ROUTE_SEND/MSG_ROUTE_RECV	Bộ định tuyến gửi/nhận thông điệp
5	ROUTE_BEGIN/ROUTE_END	Bộ định tuyến bắt đầu/kết thúc
6	NODE_BEGIN/NODE_END	Nút bắt đầu/kết thúc
7	NODE_CREATION/NODE_REMOVAL	Tạo/xóa nút
8	NODE_TIMER	Bộ đinh thời nút
9	MSG_DROPPED	Hủy thông điệp
10	MONITORING_LLINK_CREATION	Giám sát tạo liên kết
11	MONITORING_LLINK_REMOVAL	Giám sát xóa liên kết
12	MONITORING_NODE_FUNCTION_CREATION	Giám sát tạo chức năng nút
13	MONITORING_NODE_FUNCTION_REMOVAL	Giám sát xóa chức năng nút
14	MONITORING_PROP_CHANGED	Giám sát thay đổi

Hình 2.7 thể hiện kết quả đọc tô pô bao gồm các nút và cung từ tập tin BRITE vào hệ thống mô phỏng. Để thực hiện việc truyền thông điệp giữa các máy chủ trong vòng tròn ảo trong Mục 2.3 của Luận án. Sau khi tiến hành khởi tạo hoàn chỉnh tô pô, phần dưới cùng Hình 2.7 thể hiện các nút ảo được tạo, các nút này di chuyển theo nguyên lý vòng tròn ảo. Các nút ảo này đại diện cho ứng dụng cung cấp tài nguyên đối với người sử dụng, các hoạt động và sự kiện diễn ra bên trong hệ phân tán thể hiện qua Hình 2.8. Theo Hình 2.8, một sự kiện diễn ra trong hệ phân tán bao gồm các trường thông điệp thể hiện qua Bảng 2.4.



Hình 2.10: Cấu trúc thông điệp trong hệ thống mô phỏng DSSim

Cấu trúc của một thông điệp di chuyển qua hệ thống bao gồm các trường mô tả theo Hình 2.10.

2.3.2. Triển khai thuật toán 4PCoDT trong hệ phân tán

Luận án trình bày chương trình mô phỏng thuật toán 4PCoDT của hệ phân tán trong Mục 2.3 của Luận án là bài toán bãi đỗ xe mô tả trong ví dụ thứ hai trong Mục 1.3.1 của Luận án. Chương trình xây dựng trên ngôn ngữ Java, hệ quản trị cơ sở dữ liệu MySQL thực thi trên hệ điều hành Windows Server 2008 nằm trong hệ thống ảo hóa VMware vSphere.

Giải pháp đề xuất của Luận án để giải quyết bài toán điều khiển cung cấp tài nguyên dùng chung như sau:

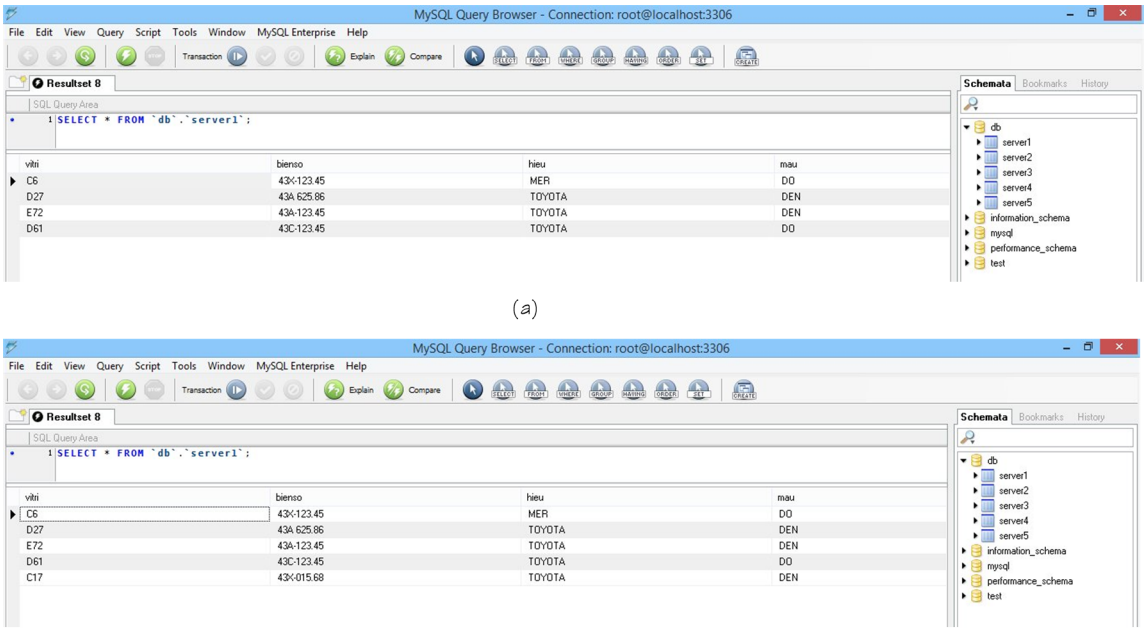
Các BV phải hợp lực với nhau để phân phối chính xác các chổ trong bãi, đặc biệt là số lượng chổ còn trống càng ít, thì vai trò của hợp lực càng quan trọng. Các bước hợp lực thể hiện như sau:

- *Bước 1*: Phối hợp, sử dụng đồng hồ lô gíc dựa trên song song hóa thuật toán Lamport để gắn giá trị cho các tiến trình (xe vào/ra) yêu cầu/giải phóng tài nguyên (VT). Sau khi đã có giá trị đồng hồ lô gíc, lập tức gán giá trị này cho tiến trình yêu cầu VT.

- *Bước 2*: Sử dụng 2 hàng đợi động để tiếp nhận và sắp xếp các giá trị đồng hồ lô gíc. Tạo ra một bảng dữ liệu tạm để cập nhật thông tin tiến trình.

- *Bước 3*: Rà soát lại các bước đã thực hiện trước khi cấp phát VT. Bước này kiểm tra lại các giá trị đồng hồ lô gíc đã cấp phát cho các tiến trình. Nếu tài nguyên nào đã cấp phát cho hai tiến trình trở lên, lập tức kiểm tra, so sánh giá trị đồng hồ lô gíc. Giá trị đồng hồ lô gíc có giá trị nhỏ hơn được ưu tiên. Các tiến trình có giá trị đồng hồ lô gíc cao hơn thì hủy cung cấp tài nguyên và thông báo đến tất cả các BV để xác nhận.

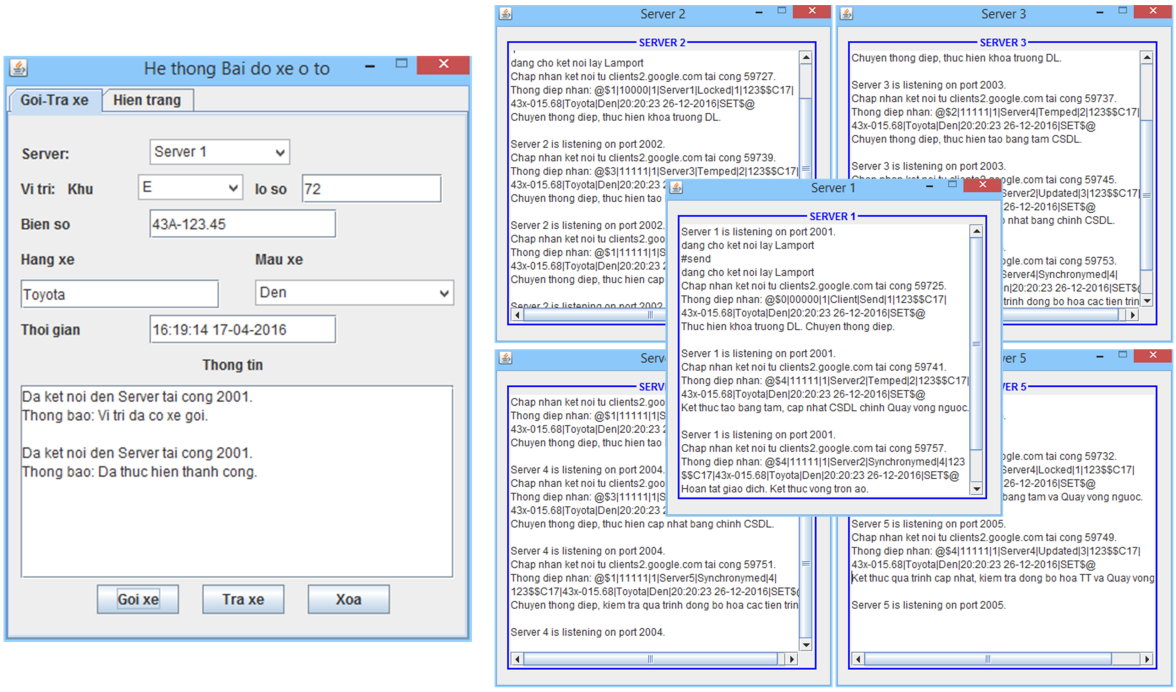
- *Bước 4*: Xử lý tuần tự các tiến trình ưu tiên sau khi sắp xếp. Khi đến giá trị đồng hồ lô gíc ở hàng đợi 2 cần được xử lý, BV nào đang chiếm giữ để cấp phát cho tiến trình phát lệnh cho phép các tiến trình yêu cầu/giải phóng tài nguyên. Sau khi hoàn thành, lưu vết thông tin toàn bộ quá trình thực hiện của các tiến trình.



Hình 2.11: Trạng thái các bảng dữ liệu trên các máy chủ

Các bước thực hiện nêu trên chính là các pha trong thuật toán 4PCoDT. Mục đích của Luận án là trình bày giải pháp đảm bảo tính gắn bó trong hệ phân tán, chương trình thực hiện trên tầng SaaS của mô hình Điện toán đám mây. Thông qua thuật toán, các chương trình được triển khai trên các máy chủ ảo nhằm cung cấp tài nguyên dùng chung đối với người sử dụng. Các chương trình này là một nút ảo thực hiện mô phỏng phần 2.5.1 thể hiện việc di chuyển theo nguyên lý vòng tròn ảo trong tô pô. Mỗi thông điệp được tạo ra và định tuyến đi trong vòng tròn là các sự kiện được tạo ra trong Bảng 2.3.

Hình 2.11a thể hiện trạng thái hiện tại các bảng dữ liệu trên các máy chủ, các trường dữ liệu thể hiện trạng thái nhất quán. Chương trình từ người sử dụng truy cập đến máy chủ bất kỳ trong hệ thống để đăng ký vị trí trong bãi đỗ thể hiện Hình 2.12a. Sau khi đăng ký hoàn tất, thông báo gửi về báo cho người sử dụng biết quá trình thực hiện đã thành công.



(a) (b)

Hình 2.12: Chương trình thể hiện thuật toán 4PCoDT

Kết quả hoạt động trên các máy chủ thể hiện trong Hình 2.12b. Khi người sử dụng bắt đầu kết nối yêu cầu tài nguyên, giá trị đồng hồ lô gíc được khởi tạo để gắn giá trị cho thông điệp và bắt đầu các pha của giao dịch. Các thông điệp dịch chuyển theo nguyên lý vòng tròn ảo để thực hiện 4PCoDT. Khi thông điệp di chuyển qua từng máy chủ để thực hiện xử lý thông điệp, các trường điều khiển được tách ra và xử lý dựa vào từng trường cụ thể. Các trường sau khi được xử lý sẽ cập nhật lại để tái thiết lập trước khi gửi cho máy chủ tiếp theo trong vòng tròn.

Kết thúc giao dịch, dữ liệu trên các máy chủ cho kết quả giống nhau được thể hiện ở Hình 2.11b, trạng thái này gọi là *nhất quán dữ liệu trong hệ phân tán.*

2.3.3. Đánh giá và nhận xét các thuật toán trong hệ phân tán

Trong quá trình nghiên cứu và thực nghiệm, Luận án đã đưa ra mô hình cung cấp tài nguyên cho hệ phân tán ảo. Đặc tính này quan trọng đối với các ứng dụng Diện toán đám mây, khi nhu cầu tài nguyên dùng chung trên đám mây không còn phụ thuộc vào hệ quản trị cơ sở dữ liệu cục bộ [28, 76, 101] mà chỉ cần mô tả cấu trúc cho phép truy vấn thông qua cơ sở dữ liệu như là một dịch vụ DBaaS (Database as a Service).

So với các thuật toán hiện tại như 2PC, 3PC, thuật toán 4PCoDT có sự cải tiến,

đó là pha kiểm tra đồng bộ hóa tiến trình dựa vào giá trị đồng hồ lô gíc. Giá trị đồng hồ lô gíc nhằm để xác định độ ưu tiên trong trật tự tổng quát các tiến trình yêu cầu tài nguyên để đảm bảo phòng, tránh bế tắc. Ngoài ra, các pha thực hiện trong thuật toán 4PCoDT cho phép *tài nguyên cấp cho tiến trình là duy nhất, đảm bảo sự gắn bó trong hệ thống và nhất quán dữ liệu trên các máy chủ ảo.* Qua bốn pha thực hiện trong vòng tròn ảo, kết quả các tiến trình rơi vào một trong hai trường hợp: được và không được cung cấp tài nguyên dùng chung. Đối với các tiến trình không được cung cấp tài nguyên dùng chung, các pha giao dịch tiến hành hủy bỏ giao dịch và thông báo cho người sử dụng, tránh trường hợp phải mất chi phí khôi phục giao dịch như thuật toán 2PC, 3PC trong [99].

Bảng 2.5: So sánh giao thức 2 pha, 3 pha và 4 pha

	2 pha	**3 pha**	**4 pha**
Loại giao dịch	2 pha	2 pha mở rộng	3 pha mở rộng
Pha bầu chọn	x	x	x
Pha tiền ủy thác	-	x	x
Pha kiểm tra đồng bộ tiến trình	-	-	x
Pha ủy thác	x	x	x
Khóa giao dịch	thấp	cao	cao
Trao đổi thông điệp	$4(n\text{-}1)$	$5(n\text{-}1)$	$6(n\text{-}1)$
Chi phí truyền thông	thấp	cao	cao
Ghi log	$2n$	$2n$	$2n$
Độ phức tạp	$O(n^2)$	$O(n^3)$	$O(n^4)$
Chi phí khôi phục giao dịch	thấp	cao	-
Hiệu năng	thấp	cao	cao
Áp dụng cho giao dịch phân tán	khó khăn	dễ dàng	dễ dàng

Bảng 2.5 thể hiện kết quả so sánh các giao dịch 2 pha, 3 pha và 4 pha. Giao dịch 4 pha có ưu điểm hơn so với giao dịch 3 pha là kiểm tra quá trình đồng bộ hóa các tiến trình diễn ra bên trong hệ phân tán để đảm bảo tính gắn bó mạnh và đạt được hiệu năng cao. Bên cạnh đó, pha kiểm tra đồng bộ hóa tiến trình phát hiện các giao dịch bị lỗi lập tức thông báo và hủy các giao dịch liên quan đến tiến trình đang quản lý nên không có chi phí khôi phục giao dịch như giao dịch 2 pha và 3 pha. Ngược lại, nhược điểm giao dịch 4 pha sử dụng nhiều pha hơn so với 2 pha và 3 pha nên có nhược điểm là lượng thông điệp trao đổi nhiều hơn dẫn đến chi phí truyền thông và ghi log cao, độ phức tạp lớn.

Tiểu kết Chương 2

Công trình nghiên cứu trong Chương 2 đã trình bày các vấn đề giải pháp cung cấp tài nguyên truyền thông trong hệ phân tán và thuật toán đảm bảo tính gắn bó trong hệ phân tán ảo. Song song hóa thuật toán Lamport trong công trình công bố số (2) nhằm gắn dấu và thiết lập trật tự tổng quát chặt chẽ các tiến trình hoạt động trong hệ phân tán. Áp dụng song song hóa thuật toán Lamport trong loại trừ tương hỗ để đảm bảo tiến trình duy nhất vào miền găng và đạt được hiệu năng cao của hệ phân tán. Thuật toán 4PCoDT trong công trình công bố số (4) đảm bảo tính gắn bó tập trung vào việc xây dựng thông điệp, cơ chế điều khiển, giám sát và định tuyến từ nguồn đến đích. Về cấu trúc thông điệp thể hiện đặc tính mới cho hệ phân tán để xác lập cơ chế điều khiển qua lại giữa các máy chủ nhằm hợp lực cung cấp tài nguyên dùng chung. Chương trình điều khiển thông điệp được phát triển trên nguyên lý vòng tròn ảo, dựa vào đó Luận án đưa ra thuật toán 4PCoDT nhằm đảm bảo tính gắn bó cho hệ phân tán thông qua môi trường truyền thông. Nhược điểm trong cung cấp tài nguyên truyền thông trong hệ phân tán dựa trên cơ chế truyền multicast là thông tin bị trùng lặp tại tập đích dẫn đến lãng phí tài nguyên truyền thông. Giải pháp xử lý trùng lặp gói tin có nhiều giải pháp được trình bày như: phương pháp tiếp cận vùng lân cận được sắp xếp (Sorted Neighbourhood Approach) của Hernández và Stolfo [44]; giải pháp thực thể (Entity Resolution) của các tác giả Elmagarmid, Ipeirotis và Verykios [29]; phương pháp tiếp cận dựa trên khoảng cách (Distance-Based Approach) của các tác giả Whang, Marmaros và Garcia-Molina [133]; chiến lược đếm trùng lặp (Duplicate Count Strategy) của các tác giả Draisbach và cộng sự [27]; ngoài ra, tác giả Zahid và Sharif trình bày giải pháp so khớp nội dung (Content Matching) [141]. Có nhiều giải pháp khác nhau để giải quyết vấn đề vừa nêu, xong Luận án chỉ tập trung vào giải pháp kỹ thuật mã mạng.

Chương 3.

GIẢI PHÁP KỸ THUẬT MÃ MẠNG TỐI ƯU ĐIỀU KHIỂN CUNG CẤP TÀI NGUYÊN TRUYỀN THÔNG TRONG HỆ PHÂN TÁN

Kỹ thuật mã mạng là giải pháp kỹ thuật mạng cho phép dữ liệu truyền được mã hóa và giải mã nhằm tăng thông lượng và tránh trùng lặp gói tin tại tập đích. Szabó và cộng sự trong [121] trình bày nghiên cứu mã mạng như là dịch vụ. Các tác giả cũng đã nêu lên hiệu quả của mã mạng đối với thông lượng, độ trễ, độ tin cậy, an ninh giữa các máy chủ trong hệ phân tán. Các tác giả cũng nhấn mạnh 3 hướng cơ bản, 3 hướng đó đã củng cố cho sự phát triển của Luận án:

1. Giải thích làm thế nào mã mạng cải thiện hiệu năng mạng.

2. Mã mạng kết hợp với SDN.

3. Hiệu năng của hệ thống sử dụng mã mạng so với hệ thống không dùng mã mạng.

Ngoài ra, Szabó và các cộng sự trình bày sự phát triển của cách mạng 5G trong nghiên cứu [122] đó là kiến trúc SDN khai thác mã mạng như là dịch vụ. Dựa vào những đặc tính và ưu điểm của kỹ thuật mã mạng, Chương 3 của Luận án tập trung vào giải pháp điều khiển tỷ lệ nguồn với mã mạng, giải pháp tối ưu truyền thông multicast với mã mạng và thuật toán xác định mã mạng tuyến tính. Các giải pháp trong Chương 3 giải quyết các nhóm bài toán giải pháp cung cấp tài nguyên truyền thông cho hệ phân tán, điều khiển tỷ lệ nguồn và tối ưu truyền thông multicast với mã mạng. Các nhóm giải pháp được triển khai phần 2 và 3 trên tầng SaaS và PaaS trong hệ thống ảo hóa trong mô hình tổng quát ở Hình 1.23.

3.1. Giải pháp điều khiển tỷ lệ nguồn với mã mạng

3.1.1. Các ràng buộc trong giải pháp cơ bản của kỹ thuật mã mạng

Theo tác giả Ahuja và cộng sự trong [2] trình bày bài toán chi phí lưu lượng như sau: Mạng $G = (\mathbb{U}, \mathbb{V})$ là mạng có hướng được xác định bởi tập $\mathbb{U} = \{S_i | i = 0..n\}$ là tập các nút S_i và tập $\mathbb{V} = \{S_{ij} | i = 0..n, j = 1..m\}$ là tập các cạnh S_{ij} có hướng. Mỗi cạnh $S_{ij} \in \mathbb{U}$ có chi phí cp_{ij} biểu thị chi phí trên một đơn vị lưu lượng trên cạnh đó.

Chúng ta giả định chi phí lưu lượng thay đổi tuyến tính với lượng lưu lượng. Chúng ta sẽ kết hợp với mỗi cạnh $S_{ij} \in \mathbb{V}$ một trọng số ts_{ij} biểu thị cho cực đại lưu lượng và cận dưới l_{ij} biểu thị lượng cực tiểu lưu lượng trên cạnh. Chúng ta kết hợp với mỗi nút $S_i \in \mathbb{U}$ một tỷ lệ tl_{ij} đại diện cho yêu cầu/đáp ứng. Nếu $tl_{ij} > 0$, nút i là nút yêu cầu; nếu $tl_{ij} < 0$, nút i là nút đáp ứng với giá trị $-tl_{ij}$; nếu $tl_{ij} = 0$, nút i là nút trung gian chuyển tiếp thông tin.

Theo tác giả Chou và cộng sự trong [17], tác giả Ho và cộng sự trong [46] trình bày mã mạng được xây dựng sử dụng cách tiếp cận của mã mạng tuyến tính ngẫu nhiên phân tán, trong đó các nút mạng hình thành các gói dữ liệu ra bằng cách kết hợp tuyến tính ngẫu nhiên của các khối các bit tương ứng trong các gói tin vào. Sự kết hợp tuyến tính tương ứng với mỗi gói có thể được xác định bởi hệ số véc tơ trong phần đầu gói, được cập nhật bằng cách áp dụng hệ số các véc tơ biến đổi tuyến tính giống như dữ liệu. Mỗi nút trung gian nhận gói tin với xác suất cao là tập các gói tin với hệ số véc tơ tuyến tính độc lập cho phép giải mã. Bít phần đầu của các hệ số véc tơ phụ thuộc vào các thông số của mã mạng được chọn để tác động bít phần đầu dựa trên hiệu năng thực thi và nó giảm dần theo kích thước của gói tin.

Mã mạng thực hiện các dòng thông tin (ký hiệu dt) cho các nút đích khác nhau của phiên multicast để chia sẻ dung lượng mạng bằng cách mã hóa cùng nhau: với mỗi phiên multicast đơn p có tỷ lệ truyền thông tl^p từ nguồn đến các điểm đích. Dòng vật lý (ký hiệu dv) thể hiện giá trị tối đa truyền thông tin trên mỗi liên kết $(k,\ l)$ theo cấu hình liên kết các thiết bị định tuyến. Đối với mỗi liên kết $(k,\ l)$, S_k là điểm nguồn và S_l là điểm đích, $k, l \in \mathbb{U}$. Theo nguyên lý của tác giả Ahuja và cộng sự trong [2], các ràng buộc về tỷ lệ dòng thông tin trên mỗi liên kết $(k,\ l)$ thể hiện qua Công thức (3.1) của Luận án như sau:

$$\sum_{l:(k,l)}^{\mathbb{U}} dt_{k,l}^{pS_l} - \sum_{l:(k,l)}^{\mathbb{U}} dt_{l,k}^{pS_k} = \begin{cases} tl^p \text{ nếu } k = S_l \\ -tl^p \text{ nếu } k = S_k \\ 0 \text{ ngược lại} \end{cases} \tag{3.1}$$

trong đó: mỗi liên kết $(k,\ l)$: $dt_{k,l}^{pS_l}$ là dòng thông tin từ S_k đến đích S_l của phiên p. Nếu là dòng thông tin yêu cầu từ S_k đến S_l nhận tỷ lệ đối với phiên p tl^p, nếu là dòng thông tin đáp ứng từ S_l đến S_k nhận tỷ lệ đối với phiên p $-tl^p$, đối với dòng thông tin là điểm nguồn S_k hoặc điểm đích S_l nhận giá trị 0.

$dv_{k,l}^p$ là ký hiệu dòng vật lý trên mỗi liên kết $(k,\ l)$ của phiên p với ràng buộc thể hiện qua Công thức (3.2) như sau:

$$dt_{k,l}^{pS_l} \leq dv_{k,l}^p, \forall S_l \in \mathbb{D}^p \tag{3.2}$$

phương trình cân bằng dòng thông tin (3.1) có hình thức tương tự như các dòng phương trình cân bằng dòng vật lý cho định tuyến của các luồng dữ liệu trong mạng, nghĩa là dòng thông tin truyền không vượt quá dòng vật lý đã được cấu hình. Bất đẳng thức (3.2) phản ánh điều kiện mã mạng liên quan tỷ lệ vật lý và thông tin được xác định theo Công thức (3.4):

$$dv_{k,l}^{p} = \max_{l \in \mathbb{D}^p} \left\{ dt_{k,l}^{pS_l} \right\} \tag{3.3}$$

Hình 1.27 mô tả mã mạng tuyến tính với biến các dòng tương ứng $(dv_{k,l}, dt_{k,l}^{S_5}, dt_{k,l}^{S_6})$. Khái niệm các gói dữ liệu được phát biểu như sau:

Véc tơ tỷ lệ f đáp ứng các các ràng buộc theo Công thức (3.1) *và* (3.2) *khi và chỉ khi tồn tại mã mạng thiết lập một kết nối multicast ở mức tùy ý gần bằng* tl^p *từ nguồn* S_0 *tới tập đích* $\mathbb{D}^p$ *và đưa vào các gói với tỷ lệ tùy ý gần bằng* $dt_{k,l}$ *trên mỗi liên kết (k,l).*

Đối với trường hợp chia sẻ của nhiều phiên trên mạng, thông lượng tối ưu đạt được khi yêu cầu một số trường hợp mã hóa qua nhiều phiên. Tuy nhiên, thiết kế mã như vậy là vấn đề phức tạp và mở rộng bài toán. Do đó, giới hạn của Luận án việc xem thực hiện mã mạng cùng các phiên, phương pháp tiếp cận gọi là mã chập (convolutional code) trình bày bởi Koetter và Medard trong [57] hoặc mã hóa phiên cục bộ. Tập các vec tơ dòng thông tin có tính khả thi được xác định bởi các kết hợp ràng buộc (3.1) và (3.2) cho mỗi phiên $p \in \mathbb{P}$ với ràng buộc trọng số liên kết được xác định theo Công thức (3.4) như sau:

$$\sum_{p}^{\mathbb{P}} dv_{k,l}^{p} \leq ts_{k,l}, \forall (k,l) \in \mathbb{V} \tag{3.4}$$

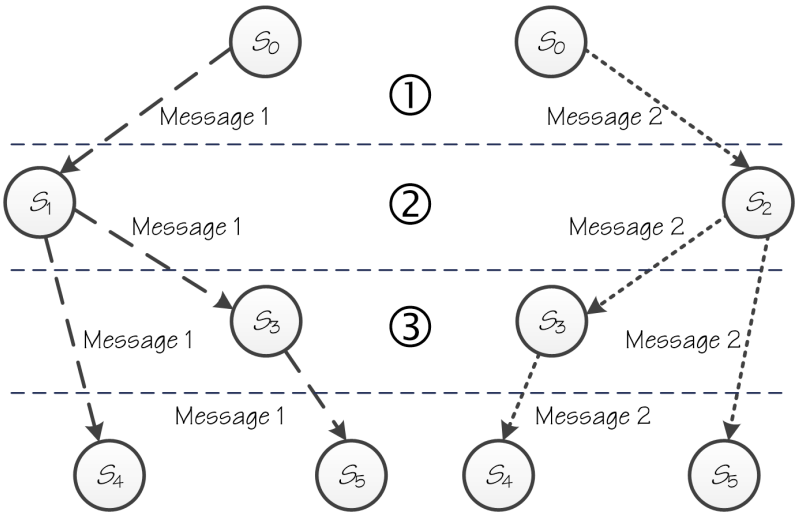
Các Công thức từ (3.1) đến Công thức (3.4) thể hiện các ràng buộc trong truyền thông tin với mã mạng multicast. Thông tin nguyên thủy được chia thành nhiều gói và truyền multicast qua các kênh truyền dựa trên các tỷ lệ khác nhau tuân thủ các ràng buộc trong truyền thông. Vì vậy, cần phải xác định tỷ lệ cực đại truyền trên từng kênh. Giải pháp để đạt được tỷ lệ cực đại trên các kênh truyền là phải xác định được tỷ lệ và điều chỉnh tỷ lệ trên từng nhánh của cây multicast.

3.1.2. Xác định các tỷ lệ và tối ưu hóa điều khiển tỷ lệ với đồ thị con

Luận án xét mạng với mã đồ thị con G^p cho mỗi phiên p, G^p có thể được xem là sự kết hợp tập $\mathbb{X}^p$ chứa các liên kết của của cây multicast bao trùm, mỗi kết nối

nguồn S_0 đến tập đích $\mathbb{D}$. Điều khiển tỷ lệ được thực hiện bằng cách điều chỉnh tỷ lệ dòng thông tin trên từng cây. Mã hóa được thực hiện trên các đoạn bao trùm của các cây khác nhau của phiên p để có các tập rời rạc truyền đến nút đích. Hình 3.1 thể hiện cây muticast được triển khai từ các mã đồ thị con thể hiện trong Hình 1.27. Theo Hình 1.27, mã hóa trên các liên kết chia sẻ cho phép cả hai cây đồng thời truyền thông tin tỷ lệ riêng ở mức cực đại.

Tương tự như định tuyến thông tin, mã đồ thị con được chọn nhiều cách khác nhau dựa trên sự kết hợp nhiều yếu tố khác nhau như: độ trễ, cung cấp tài nguyên, các cam kết của nhà cung cấp dịch vụ mạng. Các thuật toán xây dựng cây multicast hoặc các kỹ thuật hiện có được áp dụng trong việc tìm kiếm nhiều đường dẫn đến mỗi đích và kết hợp tập thích nghi của các đường dẫn đã tạo cây.



Hình 3.1: Cây multicast

Để đơn giản hóa các ký hiệu, Luận án xem xét trường hợp các đoạn bao trùm của cây khác nhau của phiên có các tập rời rạc truyền đến nút đích. Do đó, mã hóa xảy ra trên tất cả các đoạn bao phủ. Trong trường hợp tổng quát hơn, mã hóa chỉ diễn ra trên một vài đoạn bao phủ. Mỗi cây ký hiệu là $C_x^p, x \in \mathbb{X}^p$ chứa tập $\mathbb{V}_x \in \mathbb{V}$ của các liên kết, trong đó xác định $\mathbb{V} \times \mathbb{X}^p$ là ma trận multicast MT^p có phần tử thứ $(v,\ x)$ được xác định bởi Công thức (3.5):

$$MT_{vx}^p = \begin{cases} 1 \text{ nếu } v \in \mathbb{V}_x \\ 0 \text{ ngược lại} \end{cases} \tag{3.5}$$

Trên mỗi cây multicast C_x^p, từ nguồn gửi các dòng thông tin đến tập đích đặc trưng bởi tỷ lệ tl_x^p, đây là tỷ lệ truyền thông giữa dòng thông tin và dòng vật lý. Với

mã mạng phiên cục bộ, tỷ lệ dòng vật lý đặc trưng dv_v^p cho mỗi phiên multicast p với liên kết v là $dv_v^p = \max_{x \in \mathbb{X}_p} \{MT_{vx}^p tl_x^p\}$. Trọng số ràng buộc liên kết ở Công thức (3.4) được xác định theo Công thức (3.6):

$$\sum_{p}^{\mathbb{P}} dv_v^p = \sum_{p}^{\mathbb{P}} \max_{x \in \mathbb{X}_p} \{MT_{vx}^p tl_x^p\} \leq ts_v, \forall v \in \mathbb{V} \tag{3.6}$$

Mỗi phiên p có hàm tiện ích $T^p\left(tl^p\right)$ khi truyền với tỷ lệ $tl^p = \sum_{x}^{\mathbb{X}_p} tl_x^p$ gói trên giây qua các mã đồ thị con. Giả sử $T^p(\cdot)$ là khả vi liên tục, tăng, và lõm ngặt cho dòng với yêu cầu tỷ lệ thay đổi. Cách tiếp cận của Luận án là chọn tỷ lệ nguồn cực đại của tl_x^p để giải quyết bài toán tổng quát theo Công thức (3.7) như sau:

$$\begin{aligned} &\max_{tl_x^p, dv_v^p} \sum_{p}^{\mathbb{P}} T^p\left(tl^p\right) \\ &\text{Áp dụng cho: } MT_{vx}^p tl_x^p \leq dv_v^p, \forall x \in \mathbb{X}_p, \forall p \in \mathbb{P} \\ &\qquad\qquad \sum_{p}^{\mathbb{P}} dv_v^p \leq ts_v, \forall v \in \mathbb{V} \end{aligned} \tag{3.7}$$

Công thức (3.7) áp dụng cho việc tính toán tỷ lệ cực đại cho các dòng thông tin multicast trong mã mạng. Dựa vào giá trị trong Công thức (3.7), điều khiển tỷ lệ trên các cây multicast sẽ được tối ưu trên mỗi phiên truyền thông tin.

Luận án trình bày mỗi phiên multicast p chia lưu lượng thành các biến $ll_x^p \geq 0$ cho mỗi cây multicast C_x^p của đồ thị con được mã, để $\sum_{x}^{\mathbb{X}_p} ll_x^p = 1$ và $tl_x^p = tl^p ll_x^p$. Chúng ta biết rằng ll_x^p điều khiển các phần của lưu lượng của phiên multicast p được gửi thông qua cây C_x^p. Thay vì giải quyết bài toán trực tiếp ở Công thức (3.6), trước tiên chúng ta xét bài toán điều khiển tỷ lệ với véc tơ chia cố định ll, từ Công thức (3.6) thay đẳng thức $tl_x^p = tl^p ll_x^p$ ta xác định Công thức (3.8) như sau:

$$\begin{aligned} &\max_{tl_x^p, dv_v^p} \sum_{p}^{\mathbb{P}} T_p\left(tl^p\right) \\ &\text{Áp dụng cho: } MT_{vx}^p tl^p ll_x^p \leq dv_v^p \\ &\qquad\qquad \sum_{v}^{\mathbb{V}} dv_v^p \leq ts_v \end{aligned} \tag{3.8}$$

Bài toán trên là hàm lồi ngặt và có nghiệm duy nhất theo tỷ lệ nguồn tl^p, cực đại bởi $T(ll)$. Bài toán ở Công thức (3.6) tương ứng với các tính toán xác định theo Công thức (3.9):

$$\begin{aligned} &\max_{ll \geq 0} T\left(ll\right) \\ &\text{Áp dụng cho: } \sum_{x}^{\mathbb{X}^p} ll_x^p = 1 \end{aligned} \tag{3.9}$$

Bài toán trên không nhất thiết phải là hàm lồi, nhưng vẫn có thể được giải quyết cho bài toán tối ưu tổng quát. Để xử lý các giá trị trên hàm hồi, xét Lagrangian (ký hiệu L) của Công thức (3.7) theo các ràng buộc cho mã mạng xác định theo Công thức (3.10):

$$L\left(ll, tn, tl, dv\right) = \sum_{p}^{\mathbb{P}} T^{p}\left(tl^{p}\right) - \sum_{v,p,x}^{\mathbb{V},\mathbb{P},\mathbb{X}_{p}} tn_{v,x}^{p}\left(MT_{vx}^{p} tl^{p} ll_{x}^{p} - dv_{v}^{p}\right) \tag{3.10}$$

Trong đó: $tn_{v,x}^{p}$ là "*giá trị tắc nghẽn*" tại liên kết v đối với cây multicast C_{x}^{p} và giá trị tối đa Lagrangian thông qua tl và dv, tn giá trị cố định được xác định cho trước, chúng ta có điều khiển tỷ lệ chung và thuật toán lập lịch phiên:

Điều khiển tỷ lệ: Tại thời điểm t, cho giá trị tắc nghẽn $tn(t)$, các nguồn S_i điều chỉnh tỷ lệ dòng tl^{p} theo tổng giá trị tắc nghẽn trên cây multicast C_{x}^{p}, tỷ lệ phiên p tại thời điểm t được xác định Công thức (3.11) như sau:

$$tl^{p}\left(t\right) = \left(T^{p}\right)^{-1}\left(\sum_{x}^{\mathbb{X}^{p}} ll_{x}^{p} \sum_{v}^{\mathbb{V}} MT_{v,x}^{p} tn_{v,x}^{p}\left(t\right)\right) \tag{3.11}$$

Công thức (3.11) tương tự như thuật toán điều khiển tắc nghẽn TCP, trong đó nguồn điều chỉnh tỷ lệ gửi theo tổng giá trị tắc nghẽn của phiên p.

Lập lịch phiên: mạng tuyến tính ngẫu nhiên hệ thống được mã hóa sử dụng gói tin mã từ phiên multicast tương tự [137]. Trên liên kết v, gửi một số lượng gói tin (ký hiệu là g) được mã hóa cho phiên p: $g_{v}\left(t\right) = \arg\max_{p \in P} \sum_{x}^{\mathbb{X}^{p}} tn_{v,x}^{p}\left(t\right)$ với trọng số ts_{v}. Để đạt được giá trị tối đa thông qua $\{dv_{v}^{p}\}$ là một chương trình tuyến tính được xác định trong Công thức (3.12) như sau:

$$dv_{v}^{p}\left(t\right) = \begin{cases} ts_{v} \\ 0 \end{cases} \tag{3.12}$$

Đặt hàm tiện ích tính lưu lượng và giá trị tắc nghẽn $H(\cdot)$: $H\left(ll, tn\right) = \max_{tl,dv} L\left(ll, tn, tl, dv\right)$ với $\sum_{p}^{\mathbb{P}} dv_{v}^{p} \leq ts_{v}$, ta xác định Công thức (3.13):

$$T\left(ll\right) = \min_{tn \geq 0} H\left(ll, tn\right) = \min_{tn \geq 0} \max_{tl,dv} L\left(ll, tn, tl, dv\right) \tag{3.13}$$

Bài toán tối ưu $\min_{tn} H\left(ll, tn\right)$ có thể giải sử dụng phương pháp subgradient, trong đó số nhân Lagrangian được điều chỉnh theo chiều ngược với subgradient.

Cập nhật giá trị tắc nghẽn: Mỗi liên kết v cập nhật giá trị tắt nghẽn theo cây multicast C_x^p xác định theo Công thức (3.14):

$$tn_{v,x}^p(t+1) = \left[tn_{v,x}^p(t) + y_t\left(MT_{vx}^p tl^p ll_x^p(tn(t)) - dv_v^p(tn(t))\right)\right]^+ \tag{3.14}$$

Trong đó: y_t là một bước kích thước vô hướng dương, "+" ký hiệu phép chiếu tập số thực không âm $\mathbb{R}^+$.

Liên kết v sẽ sử dụng trọng số dv_v^p để truyền gói tin được mã cho phiên multicast p, phương trình (3.14) chỉ ra rằng nếu yêu cầu ma trận $MT_{vx}^p tl_x^p$ cho trọng số ảo tại liên kết v đối với dòng thông tin của cây multicast C_x^p vượt quá trọng số vật lý được gán dv_v^p, giá trị $tn_{v,x}^p$ sẽ tăng lên và giảm các tham số khác. Do đó, phương trình (3.14) là phương trình phân tán và được thực hiện tại các liên kết riêng sử dụng thông tin cục bộ.

Các thuật toán điều khiển tỷ lệ trên (3.11) và (3.14) thực hiện với giả thiết véc tơ chia lưu lượng ll chứa các hằng số. Luân án nghiên cứu bằng cách nào để điều chỉnh lưu lượng ll_x^p để giải quyết bài toán ở (3.7) gọi là cây thích nghi. Giả sử rằng xử lý trên cây thích nghi là chậm hơn nhiều so với giá trị tối thiểu trong $H(ll,tn)$ thông qua giá trị tắc nghẽn tn có thể được xem như là tức thời.

Lý thuyết điểm cân bằng được đề xuất bởi Wardrop trong nghiên cứu [19, 132] được áp dụng cho giải pháp về khái niệm của mô hình mạng khi mô hình hóa mạng truyền thông và viễn thông với tình trạng tắc nghẽn. Khái niệm này giả định rằng người lập lịch trình chọn một định tuyến để giảm thời gian hoặc chi phí phát sinh trong định tuyến. Giả định này tiếp nhận mô tả toán học và các thuật toán hiệu quả cho việc tính toán điểm cân bằng có sẵn. Vì vậy, các nhà quy hoạch đã sử dụng khái niệm này để đánh giá các dự án, tối ưu hóa các chi phí sử dụng, nhu cầu tính toán và nhiều ứng dụng phát sinh từ thành phần mở rộng của mô hình cơ bản.

Tối ưu véc tơ chia lưu lượng tập trung vào trạng thái cân bằng tương tự như nguyên lý điểm cân bằng Wardrop, nghiên cứu điều khiển tỷ lệ lưu lượng trong [134] trình bày tổng giá trị mỗi phiên multicast trong tất cả các cây multicast được sử dụng là như nhau. Các nghiên cứu trong [38] trình bày cập nhật các véc tơ chia lưu lượng tiếp cận trạng thái cân bằng. Tại giai đoạn z, cho véc tơ chia lưu lượng $ll(z)$, giả sử $tn(z) \in \arg\min_{tn} H(ll(z), tn)$ là giải pháp tối ưu cho bài toán ở (3.8) và để $x^p(z) = \arg\min_{x}^{\mathbb{X}} \sum_v tn_{v,x}^p(z)$.

Cây thích nghi: mỗi nút nguồn S_i cập nhật ll_x^p xác định theo Công thức (3.15):

$$ll_x^p(z+1) = ll_x^p(z) + w_x^p(z) \tag{3.15}$$

với giá trị w được xác định theo Công thức (3.16):

$$w_x^p(z) = \begin{cases} -\min\left\{ll_x^p, bk_z\left(\sum_v^{\mathbb{V}} tn_{v,x}^p(z) - \sum_v^{\mathbb{V}} tn_{v,x^p(z)}^p(z)\right)\right\}, \text{ nếu } x \neq x^p(z) \\ -\sum_{x\neq x^p(z)}^{\mathbb{X}_p} w_x^p(z) \text{ nếu } x = x^p(z) \end{cases} \tag{3.16}$$

Trong đó: bk là kích thước bước vô hướng dương. Hai ràng buộc giá trị w xác định theo Công thức (3.17) và (3.18) như sau:

$$\sum_x^{\mathbb{X}_p} w_x^p(z) = 0, \tag{3.17}$$

$$\sum_x^{\mathbb{X}_p} w_x^p \sum_v^{\mathbb{V}} tn_{v,x}^p \leq 0 \tag{3.18}$$

Trong đó: $\sum_x^{\mathbb{X}_p} w_x^p \sum_v^{\mathbb{V}} tn_{v,x}^p = 0$ khi và chỉ khi $w_x^p(z) = 0$, yêu cầu cân bằng lưu lượng của vế trên Công thức (3.16) được xác định theo Công thức (3.19):

$$ll_x^p(z)\left(\sum_v^{\mathbb{V}} tn_{v,x}^p(z) - \sum_v^{\mathbb{V}} tn_{v,x^p(z)}^p(z)\right) = 0 \tag{3.19}$$

Dẳng thức trong Công thức (3.19) là trạng thái cân bằng cho phép tối ưu véc tơ chia lưu lượng. Đạt được trạng thái này, việc truyền multicast kết hợp với mã mạng sẽ cân bằng giữa dòng vật lý cho trước và là phép tính tối ưu trong việc điều khiển các tỷ lệ nguồn. Đây là hàm mục tiêu cho phép cân bằng lưu lượng từ nguồn đến đích, thông qua giải pháp này kết hợp với thiết kế tô pô tối ưu sẽ đạt được tỷ lệ nhận tại tập đích cực đại và tránh trùng lặp thông tin trong quá trình truyền đến tập đích. Giải pháp tối ưu truyền thông đạt thông lượng cực đại tại tập đích của Luận án dựa trên giải pháp rút gọn cây và tính lưu lượng cực đại dòng multicast dựa trên cây.

Mô phỏng thực thi kỹ thuật mã mạng trong các ứng dụng: thực thi các phép toán qua trường hữu hạn Galois Field GF(nt), trong đó nt là số nguyên tố và GF(2^m); véc tơ mã hóa và giải mã cho các mạng tuyến tính (nghịch đảo ma trận trên các trường hữu hạn); giải mã gói cho mã mạng tuyến tính: đưa ra tập các gói mã tuyến tính tái cấu trúc các gói nguyên thủy không mã. Trong mô phỏng thực thi mã mạng tuyến tính của Luận án thông qua trường hữu hạn F2^8(m=8). Hình 3.2 mô tả gói tin nguyên thủy được chia thành nhiều gói tin. Các gói tin chia thành nhiều khối con và mã hóa trước khi truyền multicast.

Thuật toán 3: *Thuật toán điều khiển tỷ lệ nguồn với mã mạng*

Input : Tập tin nguồn

Output: Tập tin nguồn được chia dưới các khối theo tỷ lệ

Khởi tạo các giá trị ban đầu;

Đọc tập tin đầu vào để mã;

Khởi tạo Server ;

Khởi tạo tập đích (số lượng client kết nối);

Chia tập tin nguồn thành các khối và mã hóa;

return Số lượng các khối đã được mã hóa;

Thực thi điều khiển tỷ lệ nguồn với mã mạng được trình bày thông qua Thuật toán 3. Yêu cầu ban đầu của mô phỏng là khởi tạo tập tin để mã, dựa vào chiều dài phần tải dữ liệu và số khối, tập tin được cắt ra nhiều gói khác nhau và mã hóa theo Hình 3.3. Sau khi kết thúc quá trình mã hóa, các gói tin sẽ truyền từ Server đến các Client đang kết nối. Dựa vào các tham số được cấu hình, các gói tin đã mã sẽ chia làm nhiều khối và truyền đi trong hệ thống. Kết quả truyền gói tin đã mã từ Server đến Client thể hiện trong Hình 3.3. Trong kết quả mô phỏng, số Client kết nối đến Server là 2 tương ứng với số nút đích trong tập đích. Mỗi gói tin được cắt thành 15 khối và truyền đến Client để tái cấu trúc gói tin. Chỉ mục được lập để tính số lần truyền gói, sau khi kết thúc mỗi gói sẽ tiếp trục truyền gói tiếp theo.

Thuật toán 4: *Thuật toán truyền tuyến tính các gói tin trong mã mạng*

Input : Gói tin đầu vào

Output: Gói tin giải mã tại tập đích

Khởi tạo tham số: số khối, chiều dài phần tải dữ liệu, hệ số chiều dài phần tải dữ liệu;

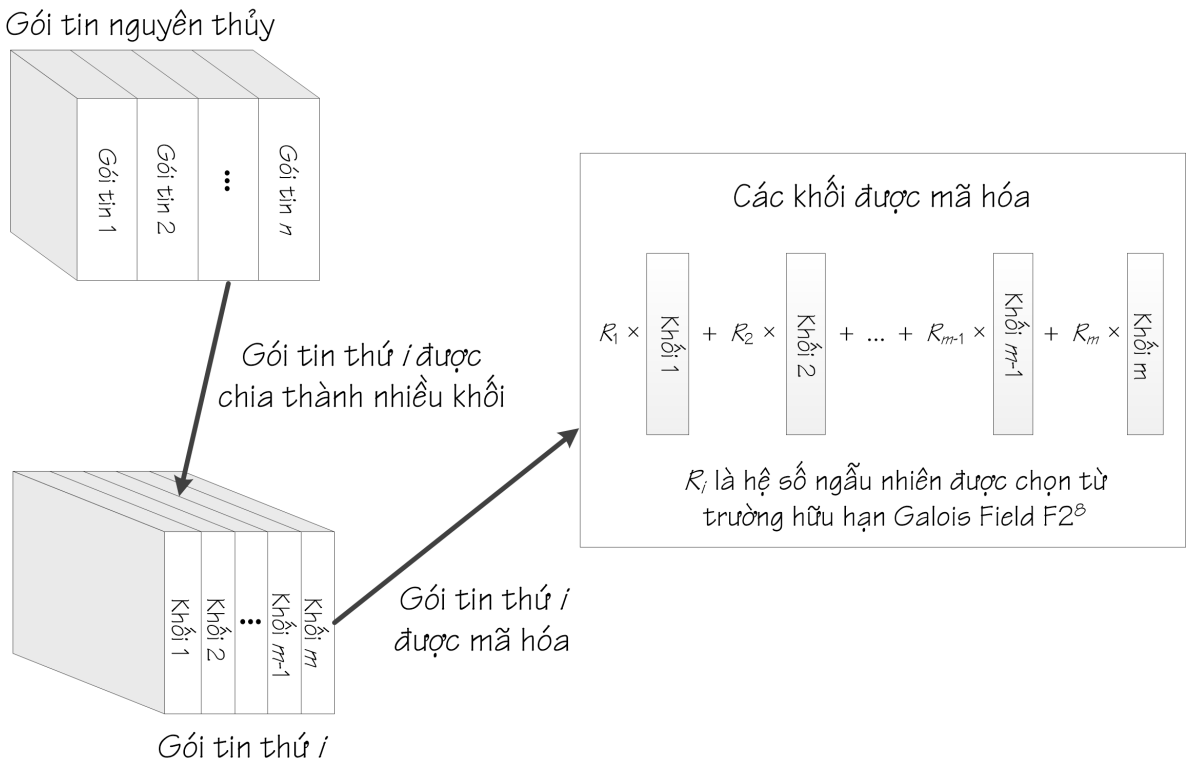
Các gói dữ liệu đầu vào;

Mã hóa các gói đầu vào và các từ mã;

Đầu ra trong hệ thống mạng: véc tơ mã gói tin, véc tơ phần tải dữ liệu gói tin

return Các gói giải mã tại tập đích;

Đối với mô phỏng xử lý bên trong truyền các gói, giải thuật cơ bản thực hiện thông qua Thuật toán 4. Dữ liệu được gửi đi được chia thành các khối được biểu diễn trong mã bởi các gói chưa mã. Các khối được kết hợp tuyến tính và truyền qua mạng. Mỗi gói được mã hóa chứa một véc tơ mã hóa mô tả cách kết hợp tuyến tính đã được



Hình 3.2: Mô tả gói tin được chia thành nhiều khối và mã hóa

xây dựng và kết hợp tuyến tính của chính nó. Các nút trung gian trong mạng nhận được các gói mã hóa tạo các kết hợp tuyến tính ngẫu nhiên khác bằng cách kết hợp ngẫu nhiên các gói đã mã. Tại đích, kết hợp tuyến tính nhận được đưa đến bộ giải mã gói tin và giải mã để tái cấu trúc thành gói tin nguyên thủy.

Đối với phần Khởi tạo tham số trong Thuật toán 4, các tham số thể hiện quá trình mã hóa gói tin: số khối thể hiện dữ liệu được chia thành khối để mã, nếu số khối này lớn quá trình chia gói sẽ nhiều và quá trình mã hóa, giải mã sẽ lớn; chiều dài phần tải dữ liệu là chiều dài của mỗi khối tính theo đơn vị byte và hệ số chiều dài phần tải dữ liệu đại diện cho véc tơ trường hữu hạn đại diện cho các khối mã hóa để truyền dữ liệu thông qua hệ thống mạng. Hệ thống mã hóa sử dụng một trường hữu hạn có kích thước 16, mỗi byte của phần tải dữ liệu được biểu diễn với hai phần tử trường, do đó số lượng các hệ số của véc tơ sẽ gấp đôi chiều dài phần tải dữ liệu.

Các gói dữ liệu đầu vào trong Thuật toán 4 là dữ liệu nguyên thủy, dữ liệu này được chia thành nhiều khối dựa vào khai báo trong phần Khởi tạo tham số. Mỗi khối được đánh dấu bằng định danh để tái cấu trúc sau khi nhận tại tập đích.

Sau khi hoàn tất giai đoạn chia khối, hệ thống bắt đầu mã hóa các gói đầu vào thông qua các khối và trường hữu hạn. Các khối này được đặt trong các véc tơ để

truyền tuyến tính thông qua hệ thống mạng. Quá trình mã hóa sẽ tạo ra hai tập véc tơ đó là véc tơ mã hóa và véc tơ phần tải dữ liệu. Gói mã hoá được tạo thành bởi véc tơ mã hóa để mô tả các gói không mã đã được kết hợp để tạo ra nó. Hệ thống tạo ra các gói tin mã hoá chỉ chứa một gói tin chưa được mã, các véc tơ mã hóa bao gồm các số 0, ngoại trừ vị trí mã là số 1.

Trong phần Đầu ra của hệ thống mạng của Thuật toán 4, nút trung gian sẽ kết hợp tuyến tính các gói bằng các hệ số ngẫu nhiên. Thuật toán 4 tạo ra nhiều kết hợp tuyến tính ngẫu nhiên với các gói không mã để đảm bảo với xác suất cao các gói tin được mã hoá có thể giải mã được.

```
run:
Khoi tao file de ma? (y/n) y
Chieu dai phan tai du lieu: 1450
So khoi: 50
Da tao 0 goi
Da tao 10 goi
Da tao 20 goi
Da tao 30 goi
Da tao 40 goi
Hoan thanh khoi tao 48 goi
So Client toi da truyen du lieu: 20
Client dang hoat dong tai dia chi 192.168.0.102
Client dang hoat dong tai dia chi 192.168.0.101
Server dang hoat dong tai cong 9864
Bo dem luong: File nguon nguon\coded1
So goi tin gui trong 1 goi ma: 15
Luong du lieu: Tong cac client truy cap 2
Luong du lieu: Dang gui den Client /192.168.0.102 khoi so 0 chi muc 0
Luong du lieu: Dang gui den Client /192.168.0.101 khoi so 0 chi muc 1
Luong du lieu: Dang gui den Client /192.168.0.102 khoi so 1 chi muc 2
Luong du lieu: Dang gui den Client /192.168.0.101 khoi so 1 chi muc 3
...
Luong du lieu: Dang gui den Client /192.168.0.102 khoi so 13 chi muc 26
Luong du lieu: Dang gui den Client /192.168.0.101 khoi so 13 chi muc 27
Luong du lieu: Dang gui den Client /192.168.0.102 khoi so 14 chi muc 28
Luong du lieu: Dang gui den Client /192.168.0.101 khoi so 14 chi muc 29
```

Hình 3.3: Chia tỷ lệ và truyền gói tin đã mã từ Server đến Client

Cuối cùng, hệ thống giải mã các kết hợp tuyến tính ngẫu nhiên bằng cách sử dụng bộ giải mã gói tin được thực hiện tại tập đích trong phần **return** của Thuật toán 4. Mỗi khi một gói tin được thêm vào bộ giải mã sẽ cố gắng giải mã càng nhiều càng tốt và tái cấu trúc các khối nó có thể giải mã.

Thông qua Thuật toán 3 và Thuật toán 4, các gói tin sẽ không bị trùng lặp tại tập đích đồng thời tăng thông lượng tại tập đích nhằm tăng tính hiệu quả đối với kỹ thuật mã mạng.

3.2. Giải pháp tối ưu truyền thông multicast với mã mạng

Nghiên cứu thiết kế và xây dựng mạng hoàn chỉnh của tác giả Kershenbaum trong [54] bao gồm nhiều khía cạnh như tính toán ma trận lưu lượng, xây dựng tô pô và quản lý. Trong đó, xây dựng tô pô với chi phí tối ưu là một trong những bài toán quan trọng của thiết kế mạng. Giải quyết bài toán tối ưu trong xây dựng tô pô, các nghiên cứu của Li và cộng sự [5], Tan và cộng sự [79] đưa ra giải pháp là giải quyết bài toán dựa trên cây. Luận án đề xuất giải pháp tối ưu truyền thông multicast với mã mạng dựa trên tối ưu từng nhánh của cây multicast theo Hình 3.1. Hướng giải quyết của Luận án là xây dựng thuật toán thêm liên kết và xóa liên kết, hai thuật toán này tạo ra cây rút gọn từ nguồn đến tập đích. Kết quả rút gọn cây từ hai thuật toán thêm liên kết và xóa liên kết phối hợp thuật toán Ford Fulkerson để tối ưu chi phí truyền thông.

3.2.1. Các yêu cầu về thông lượng và xây dựng tô pô mạng

Thông lượng cực đại của một mạng $G = (\mathbb{U}, \mathbb{V})$ có chứa phiên truyền đơn p ký hiệu là $thl(\mathbb{U})$. Luận án sẽ so sánh $thl(\mathbb{U})$ với các tham số khác đã được xác định như: *gói, độ bền* và *kết nối* để mô tả các kết nối hoặc trọng số của một mạng truyền thông. Nghiên cứu của luận án tập trung so sánh các thông số tương ứng cho truyền unicast, truyền multicast và mã mạng.

Gói đề cập đến các tính toán số lượng gói tin trên thời gian theo cặp cạnh rời rạc các cây con của G, số lượng gói của một mạng truyền thông $\mathbb{U}$ được ký hiệu là $g(\mathbb{U})$, và bằng với thông lượng cực đại mà chưa được mã.

Độ bền là một loại kết nối vùng của mạng được ký hiệu là $db(\mathbb{U})$, đó là tỷ lệ cực tiểu thể hiện qua Công thức (3.20):

$$db(U) = \frac{|\mathbb{V}_{tp}|}{(tp-1)} \tag{3.20}$$

trong đó tp là số của các thành phần truyền thông nhóm, $\mathbb{V}_{tp}$ là tập các liên kết liên thành phần và yêu cầu phải có ít nhất một nút nguồn hoặc đích trong mỗi thành phần.

Kết nối được ký hiệu là $kn(\mathbb{U})$ đề cập đến các cạnh kết nối giữa một cặp nút trong truyền thông nhóm ký hiệu là $NM\{S_0, S_j\} \subseteq \mathbb{U}$. Nghiên cứu của Luận án tập trung vào việc định tuyến phân chia và định tuyến tích hợp. Đối với định tuyến tích hợp, tất cả các trọng số liên kết và tỷ lệ lưu lượng có giá trị nguyên. Đối với định tuyến phân chia, trọng số liên kết có thể được phân chia theo cả hai hướng, và lưu lượng có

thể được phân chia và sáp nhập ở quy mô tùy ý.

Trong một mạng với một phiên unicast $\mathbb{U}$ có các thành phần $\{G, ts, NM\}$, số lượng gói $g(\mathbb{U})$ trở thành số đường dẫn cạnh rời rạc $S_0 \to S_{\mathbb{U}}$. Thông lượng $thl(\mathbb{U})$ là tỷ lệ lưu lượng thông tin cực đại có thể đạt được trong việc truyền tải $S_0 \to S_{\mathbb{U}}$. Độ bền $db(\mathbb{U})$ bây giờ được cực tiểu trong tất cả các tập cắt để tách nguồn S_0 và đích $S_{\mathbb{U}}$.

Trường hợp truyền unicast với tỷ lệ $tl_{S_0,S_{\mathbb{U}}}$ từ nút nguồn S_0 đến nút đích $S_{\mathbb{U}}$, lưu lượng unicast (ký hiệu $du(\cdot)$) vào một nút phải bằng lưu lượng unicast này ra khỏi nút này, trừ khi nút này là nguồn hoặc đích. Đối với truyền unicast trong một mạng $\mathbb{U}$, các tham số cân bằng thể hiện qua Công thức (3.21):

$$g\left(\mathbb{U}\right) = thl(\mathbb{U}) = db(\mathbb{U}) = kn(\mathbb{U}) \tag{3.21}$$

Trường hợp mã mạng truyền multicast, bài toán xây dựng tối ưu tô pô dựa trên cây multicast. Cây Steiner cơ bản truyền multicast với nút trung gian $S_t = \{S_{t,0}, S_{t,1}, ..., S_{t,\mathbb{U}}\}$ là một sự kết hợp đặc biệt trong truyền unicast mạng $\mathbb{U}$. Sự khác biệt giữa cây Steiner multicast cơ bản với nút trung gian S_t truyền unicasts mạng $\mathbb{U}$ từ nút $S_{t,0}$ đến mỗi nút trong $\{S_{t,1}, ..., S_{t,\mathbb{U}}\}$ đó là: trong cách truyền trước đây tài nguyên sử dụng của mỗi liên kết (i, j) là cực đại của dòng thông tin $\max\limits_{l=1..\mathbb{U}} du_{S_{i,j}}^{\left(S_{t,0},S_{t,l}\right)}$; trong khi ở cách truyền sau này, tài nguyên sử dụng của mỗi liên kết (i, j) là tổng của $\sum\limits_{l=1}^{\mathbb{U}} du_{S_{i,j}}^{\left(S_{t,0},S_{t,l}\right)}$. Sự khác biệt này là tính hiệu quả của cây Steiner multicast cơ bản trong việc sử dụng nguồn tài nguyên truyền thông. Bảo toàn lưu lượng truyền cây multicast cơ bản được xác định theo Công thức (3.22a), (3.22b) và (3.22c) như sau:

$$\sum_{\{j:(i,j)\mathbb{V}\}}^{\mathbb{U}} du_{S_{i,j}}^{\left(S_{t,0},S_{t,l}\right)} - \sum_{\{j:(i,j)\mathbb{V}\}}^{\mathbb{U}} du_{S_{j,i}}^{\left(S_{t,0},S_{t,l}\right)} = \begin{cases} -tl^t \text{ nếu } i = S_{t,l}, \\ tl^t \text{ nếu } i = S_{t,0}, \\ 0 \text{ ngược lại} \end{cases} \tag{3.22a}$$

$$\forall i \in \mathbb{U}, l \in \{1, ..., \mathbb{U}\},$$

$$\sum_{\{j:(i,j)\mathbb{V}\}}^{\mathbb{U}} Q\left(du_{S_{i,j}}^{\left(S_{t,0},S_{t,1}\right)}\right) \leq 1, \forall i \in \mathbb{U}, \tag{3.22b}$$

$$\sum_{\{j:(i,j)\mathbb{V}\}}^{\mathbb{U}} Q\left(du_{S_{j,i}}^{\left(S_{t,0},S_{t,1}\right)}\right) \leq 1, \forall i \in \mathbb{U} \tag{3.22c}$$

Trong đó, Công thức (3.22a) biểu thị bảo toàn lưu lượng multicast giữa cặp kết nối (i,j), nếu là dòng thông tin yêu cầu sẽ nhận tỷ lệ tl^t, dòng thông tin đáp ứng sẽ nhận tỷ lệ $-tl^t$.Trong Công thức (3.22b) và (3.22c) có hàm Q nhận giá trị 1 khi giá trị các biến bên trong lớn hơn 0, ngược lại hàm Q nhận giá trị 0.

Khi áp dụng mã mạng, bài toán thiết lập truyền multicast với tập nút S_t và tỷ lệ lưu lượng tl_{S_t} là:

1. Xác định các đồ thị con trong mạng hiện tại.

2. Xác định mã hóa để sử dụng trong đồ thị con.

Do đó, truyền multicast là một sự kết hợp đặc biệt của truyền unicast mạng $\mathbb{U}$. Tuy nhiên, khác với trường hợp của cây Steiner multicast cơ bản, có thể có nhiều đường định tuyến một thông điệp đồng thời cho mỗi phiên truyền unicast từ nguồn $S_{t,0}$ đến đích (không có ràng buộc (3.22b) và (3.22c)).

Định tuyến truyền multicast mã mạng cơ bản cũng được ứng dụng để cân bằng tải mạng. Trường hợp truyền multicast, mục đích của việc áp dụng định tuyến mã mạng cơ bản trình bày trong nghiên cứu [113] thay cho định tuyến cây Steiner cơ bản để đạt được định tuyến tối ưu và giảm tiêu thụ băng thông của mỗi kết nối. Do đó, giải pháp định tuyến mã mạng giảm tiêu thụ tài nguyên truyền thông trong một mạng.

Tương tự truyền thông tin trong cây Steiner multicast cơ bản, luuw lượng thông tin của mỗi liên kết (i,j) là giá trị cực đại một trong số lưu lượng: $\max\limits_{l\in\mathbb{U}} \sum\limits_{\{j:(i,j)\}}^{\mathbb{U}} dt_{S_{i,j}}^{(S_{t,0},S_{t,l})}$ thay cho tổng của chúng. Công thức (3.21) sử dụng đối với truyền unicast, đối với truyền multicast trong một mạng $\mathbb{U}$ với số kênh truyền là 2, các tham số từ Công thức (3.21) xác định qua Công thức (3.23):

$$1/2kn(\mathbb{U}) \leq g(\mathbb{U}) \leq thl(\mathbb{U}) \leq db(\mathbb{U}) \leq kn(\mathbb{U}) \tag{3.23}$$

Gọi ký hiệu $cp_{S_{i,j}}$ là chi phí cho mỗi đơn vị lưu lượng trên liên kết (i,j). Cách tính chi phí cực tiểu truyền multicast với nút S_t thể hiện qua Công thức (3.24) và (3.25) như sau:

Tính giá trị cực tiểu: $\sum\limits_{v=(i,j)}^{\mathbb{V}} cp_{S_{i,j}} \cdot ts_{S_{i,j}}$

Áp dụng cho:

$$\sum_{\{j:(i,j)\}}^{\mathbb{U}} dt_{S_{i,j}}^{(S_{t,0},S_{t,l})} - \sum_{\{j:(i,j)\}}^{\mathbb{U}} dt_{S_{j,i}}^{(S_{t,0},S_{t,l})} = \begin{cases} -tl^t \text{ nếu } i = S_{t,l}, \\ tl^t \text{ nếu } i = S_{t,0}, \\ 0 \text{ ngược lại} \end{cases} \tag{3.24}$$
$$\forall i,j \in \mathbb{U}, l \in \{1,...,\mathbb{U}\}.$$

$$ts_{S_{i,j}} \geq dt_{S_{i,j}}^{(S_{t,0},S_{t,l})}, \forall (i,j) \in \mathbb{V}, l \in \{1,...,\mathbb{U}\}. \tag{3.25}$$

Công thức (3.24) và (3.25) là bài toán tuyến tính với các độ phức tạp thời gian của thuật toán là đa thức để có được giải pháp tối ưu. Trong thuật toán xây dựng tô pô, Luận án quan tâm khoảng cách $kc_{i,j}$ cũng như $cp_{i,j}$ và xây dựng truyền multicast với chi phí cực tiểu cho từng yêu cầu truyền multicast.

Trong mạng G tổng lưu lượng truyền unicast và lưu lượng truyền multicast trên một liên kết $S_{i,j}$ nhỏ hơn hoặc bằng $ts_{S_{i,j}}$. Ràng buộc này được xác định qua Công thức (3.26) như sau:

$$\sum_{i_1}^{\mathbb{U}} \sum_{i_2, i_2 \neq i_1}^{\mathbb{U}} du_{S_{i,j}}^{(S_{i_1}, S_{i_2})} + \sum_{t}^{\mathbb{U}} \max_{l \in \{1, \cdots, \mathbb{U}_t - 1\}} dt_{S_{i,j}}^{(S_{t,0}, S_{t,l})} \leq ts_{S_{i,j}} \quad \forall (i,j) \in \mathbb{V} \tag{3.26}$$

Số hạng đầu tiên ở phía bên trái của Công thức (3.26) là tổng số của lưu lượng truyền unicast trên liên kết $S_{i,j}$ và số hạng thứ hai là tổng lưu lượng truyền mã mạng multicast cơ bản trên liên kết $S_{i,j}$. Trường hợp truyền multicast thứ t, tổng lượng lưu lượng trên liên kết $S_{i,j}$ là giá trị cực đại một trong số dòng truyền unicast mạng $\mathbb{U}$, có nghĩa là, tính giá trị $\max_{l \in \{1, \cdots, \mathbb{U}\}} dt_{S_{i,j}}^{(S_{t,0}, S_{t,l})}$ thay cho tổng của lưu lượng truyền unicast mạng $\mathbb{U}$ trên liên kết $S_{i,j}$.

Bài toán xây dựng tô pô của Luận án được xây dựng như sau:

1. Số nút $\mathbb{U}$ và ma trận khoảng cách $(kc_{i,j})_{\mathbb{U} \times \mathbb{U}}$
2. Ma trận yêu cầu unicast $(yc_{i,j})_{\mathbb{U} \times \mathbb{U}}$
3. Tập nút $\{S_{t,0}, S_{t,1}, ..., S_{t,\mathbb{U}}\}$ và tỷ lệ lưu lượng tl^p của yêu cầu multicast phiên p
4. Trọng số $ts_1, \ldots, ts_k$, chi phí cố định các loại đường truyền khác nhau
5. k-nút kết nối
6. Tính giá trị tối đa liên kết sử dụng $e_{\max}$.

Tính giá trị cực tiểu ma trận khoảng cách được xác định theo Công thức (3.27) như sau:

$$\sum_{i=1}^{\mathbb{U}} \sum_{j=i+1}^{\mathbb{U}} kc_{i,j} = \sum_{i=1}^{\mathbb{U}} \sum_{j=i+1}^{\mathbb{U}} \sum_{t=1}^{K} S_{i,j}^{t} \left(cp_t + kc_{i,j} \cdot ts_t\right) \tag{3.27}$$

qua các biến:

$$S_{i,j}^{1}, ..., S_{i,j}^{K} \in \mathbb{V} : 1 \leq i \leq \mathbb{U}, i+1 \leq j \leq \mathbb{U}$$

$$du_{S_{i,j}}^{(S_0,S_{\mathbb{U}})} \geq 0 : 1 \leq i,j,S_0,S_{\mathbb{U}} \leq \mathbb{U}, (i \neq j, S_0 \neq S_{\mathbb{U}})$$
$$dt_{S_{i,j}}^{(S_{t,0},S_{t,l})} \geq 0 : t,l,i,j \in \{1,...,\mathbb{U}\}, (i \neq j, S_0 \neq S_{\mathbb{U}}).$$

Áp dụng cho:

1. Ràng buộc bảo toàn lưu lượng unicast xác định theo Công thức (3.28) sau:

$$\sum_{j=1,j\neq i}^{\mathbb{U}} du_{i,j}^{(S_0,S_{\mathbb{U}})} - \sum_{j=1,j\neq i}^{\mathbb{U}} du_{j,i}^{(S_0,S_{\mathbb{U}})} = \begin{cases} -tl^{S_0,S_{\mathbb{U}}} \text{ nếu } i = S_0, \\ tl^{S_0,S_{\mathbb{U}}} \text{ nếu } i = S_{\mathbb{U}}, \\ 0 \text{ ngược lại} \end{cases} \quad (3.28)$$
$$\forall i,j \geq 1.$$

2. Ràng buộc bảo toàn lưu lượng multicast xác định theo Công thức (3.29) sau:

$$\sum_{j=1,j\neq i}^{\mathbb{U}} dt_{S_{i,j}}^{(S_{t,0},S_{t,l})} - \sum_{j=1,j\neq i}^{\mathbb{U}} dt_{S_{j,i}}^{(S_{t,0},S_{t,l})} = \begin{cases} -tl^t \text{ nếu } i = S_{t,l}, \\ tl^t \text{ nếu } i = S_{t,0}, \\ 0 \text{ ngược lại} \end{cases} \quad (3.29)$$
$$\forall t,l,i,j \in \mathbb{U}.$$

3. Ràng buộc sử dụng liên kết (i,j) dựa theo Công thức (3.26) ta xác định theo Công thức (3.30) sau:

$$du_{S_{i,j}} = \sum_{i_1=1}^{\mathbb{U}} \sum_{i_2=1,i_2\neq i_1}^{\mathbb{U}} du_{S_{i,j}}^{(S_{i_1},S_{i_2})} + \sum_{t=1}^{\mathbb{U}} \max_{l\in\{1,\cdots,|S_t|-1\}} dt_{S_{i,j}}^{(S_{t,0},S_{t,l})} \leq e_{\max} \cdot ts_{S_{i,j}}, \quad (3.30)$$
$$\forall t,l,i,j \in \mathbb{U}, (i \neq j, S_0 \neq S_{\mathbb{U}}).$$

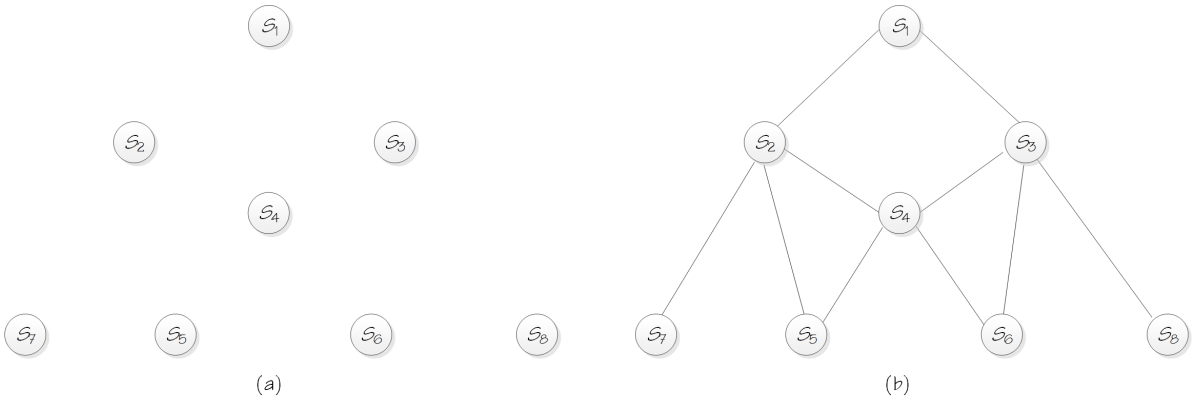
So với vấn đề xây dựng tô pô truyền thống, có thêm một ràng buộc bảo toàn lưu lượng truyền mã mạng multicast cơ bản. Khi so sánh với bài toán bảo toàn lưu lượng, ràng buộc trong Công thức (3.22a) có thêm số hạng mô tả đặc tính của mã mạng.

3.2.2. Các kỹ thuật xử lý dòng thông tin

Hệ thống các nút mạng cơ bản ban đầu thể hiện qua Hình 3.4. Trạng thái thứ nhất theo Hình 3.4.a, mạng bao gồm các nút rời rạc và chưa có liên kết. Trạng thái thứ nhất theo Hình 3.4.b, mạng bao gồm các nút liên kết hoàn chỉnh. Đối với truyền unicast, thông tin được đinh tuyến tối ưu theo thuật toán cài đặt sẵn.

Một trong những đặc điểm của hệ thống mạng truyền unicast là mạng có thể thay đổi động. Mô hình hóa mạng động bằng cách thêm/xóa các luật phối hợp giữa các nút; do đó, việc xóa một nút được mô hình hóa bằng cách xóa tất cả các luật phối hợp liên quan đến nút này. Đối với yêu cầu thêm/xóa các nút với các luật phối hợp được

dễ dàng nhận biết được với giả định rằng tất cả các nút được thiết lập ngay từ đầu và chỉ có luật phối hợp được thay đổi. Hoạt động thay đổi tô pô mạng truyền unicast diễn ra như sau:



Hình 3.4: Tô pô các nút mạng ban đầu

- AddLink(*i, j, luật, id*): thêm liên kết với luật phối hợp từ nút *j* đến nút *i*. Trong đó: *id* là định danh duy nhất cho một cặp nút.
- DeleteLink(*i, j, id*): xóa các luật phối hợp giữa các nút *id, i* và *j*.

Truyền unicast giữa hai điểm và vấn đề triển khai tô pô ban đầu với hai thuật toán thêm/xóa liên kết để tạo ra tô pô hoạt động truyền thông điệp giữa các nút mang tính chất cơ bản vừa nêu. Luận án trình bày giải pháp tối ưu tô pô, giải pháp này nhằm rút gọn cây với chi phí cực tiểu.

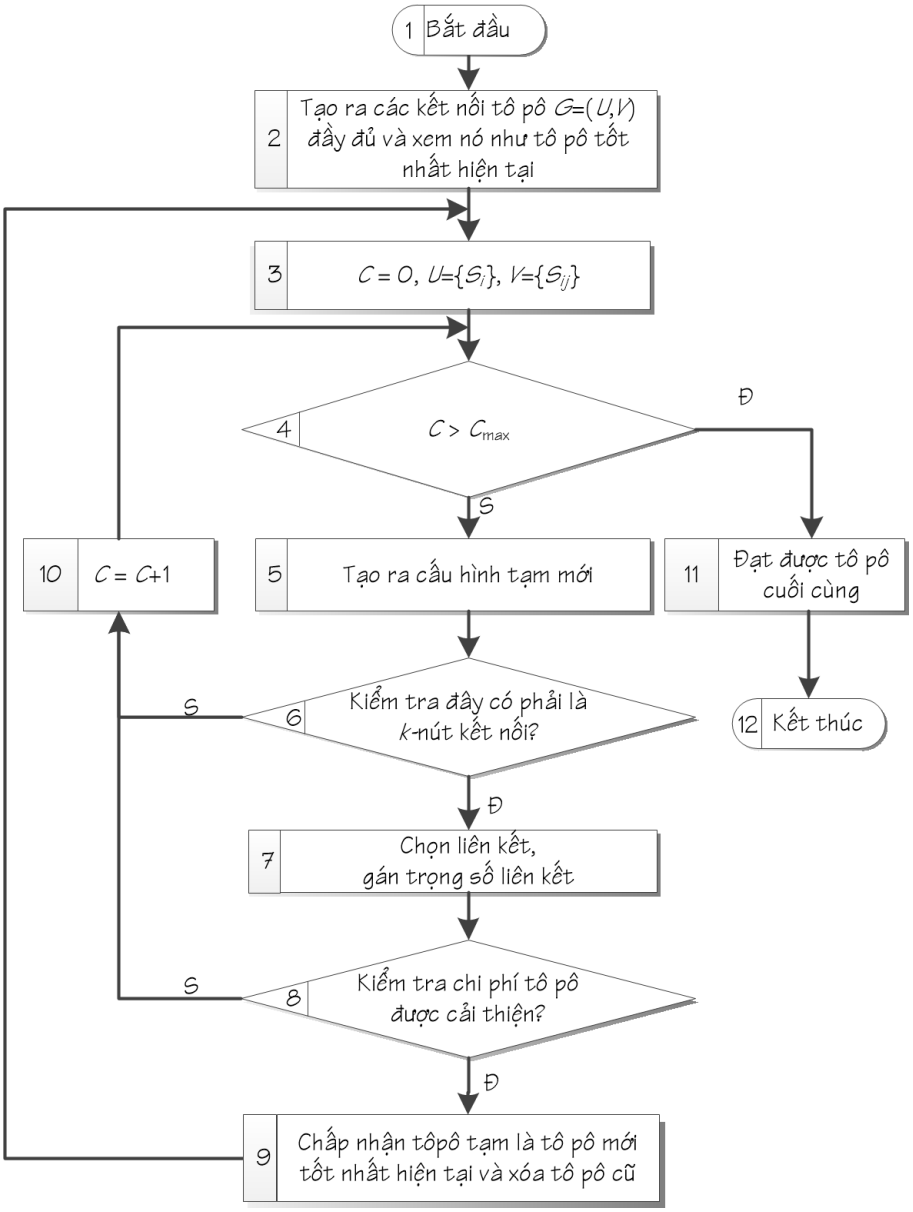
Trong các hệ phân tán phức tạp, các nút mạng phối hợp trao đổi thông điệp qua lại nên truyền unicast có nhiều hạn chế. Trên cơ sở đó, triển khai truyền multicast cho hệ phân tán trở nên cần thiết và là điều kiện để phát triển ứng dụng lớn. Luận án đề xuất hai thuật toán thêm/xóa liên kết truyền multicast và tối ưu cục bộ. Kết quả của hai thuật toán tạo ra cây multicast rút gọn theo ràng buộc k-nút kết nối. Sau khi hình thành tô pô con kết nối multicast sẽ kết hợp với mã mạng để tập đích nhận được các gói tin truyền với thông lượng tối ưu.

Hiện tại, không có thuật toán thời gian đa thức có sẵn đạt được giải pháp tối ưu của bài toán tối ưu NP-khó. Trong phần này, Luận án đưa ra hai thuật toán cho bài toán xây dựng tô pô này: *thuật toán xóa liên kết và thuật toán thêm liên kết trong cây multicast*.

Hai thuật toán đề xuất của Luận án đều bao gồm hai pha, bắt đầu tạo tô pô và tiến trình tối ưu cục bộ: Trong pha đầu tiên của thuật toán xóa liên kết, bằng cách

xóa từng liên kết từ tô pô kết nối đầy đủ cho đến khi không còn kết nối nào có thể bị xóa nữa, tô pô kết nối k-nút với chi phí thấp được tạo. Trong pha đầu tiên của thuật toán thêm liên kết, bằng cách thêm các liên kết một bởi một tô pô nguyên thủy không có liên kết cho đến khi không còn liên kết nào nữa được thêm vào, tô pô k-nút kết nối với chi phí thấp được tạo. Trong pha thứ hai của cả hai thuật toán, hoán đổi các liên kết và thực hiện vòng lặp cục bộ để cải tiến tô pô ban đầu.

3.2.2.1. Đề xuất thuật toán xóa liên kết trong cây multicast



Hình 3.5: Pha đầu trong thuật toán xóa liên kết

Mục đích của pha đầu là để tạo ra một tô pô k-nút kết nối có chi phí thấp, sơ đồ của pha đầu được thể hiện trong Hình 3.5. Theo thuật toán trong Hình 3.5, ở khối 2

là các bước tạo ra các tô pô đầy đủ kết nối và xem nó như tô pô tốt nhất hiện tại. Sau đó, có được một cấu hình tạm bằng cách xóa một liên kết cụ thể trong cấu hình hiện tại được xử lý ở các khối 6, 7 và 8. Nếu cấu hình tạm này đáp ứng, một tô pô khả thi mới với chi phí thấp hơn có thể đạt được. Chấp nhận tô pô có tính khả thi mới này là tô pô tốt nhất hiện tại mới, loại bỏ cái cũ thể hiện trong khối 9. Tham số C trong khối 3 là tham số bộ đếm sử dụng để đếm số lần thất bại liên tục, trở lại bằng không. Nếu cấu hình tạm thời này không đáp ứng tất cả những điều kiện, loại bỏ nó và tăng C lên một được thực hiện ở khối 10. Nếu giá trị của C vượt quá một giá trị nhất định $C_{max} = [\mathbb{U} \cdot k/2]$, kết thúc thuật toán và tô pô tốt nhất hiện tại là tô pô cuối cùng thể hiện ở khối 11. Ngược lại, đạt được một cấu hình tạm khác và kiểm tra nó. Bằng cách này, xóa liên kết được thực hiện vòng lặp cho đến khi không còn liên kết bị xóa.

Để thực hiện thuật toán xóa liên kết trên cây multicast, xác định một ma trận $MT_{i,j}$ trên mỗi liên kết (i, j) bởi giá trị $MT_{i,j} = kc_{i,j}/\left(cp_{i,j} + cp_{j,i}\right)$. Tiến trình thực hiện thuật toán xóa liên kết trên cây multicast bao gồm các bước cụ thể như sau:

- *Bước 1:* Chọn số nút i từ nút ngẫu nhiên 1 đến $\mathbb{U}$ và tạo ra các cấu hình kết nối đầy đủ. Sau đó, chọn định tuyến cho từng yêu cầu và xác định trọng số liên kết. Xem kết quả tô pô này là tô pô tốt nhất hiện tại.
- *Bước 2:* Thiết lập bộ đếm tham số C về không và khởi tạo $\mathbb{V}$, chứa các liên kết ứng viên để xóa, với tập chứa tất cả các liên kết trong tô pô tốt nhất hiện tại.
- *Bước 3:* Kiểm tra xem giá trị của $C > C_{max} = [\mathbb{U} \cdot k/2]$. Nếu đúng, đi đến bước 7.
- *Bước 4:* Từ $\mathbb{V}$, chọn liên kết v có giá trị ma trận hiệu suất là lớn nhất. Chứa cấu hình tạm bằng cách loại bỏ liên kết v từ cấu hình hiện tại. Kiểm tra xem cấu hình tạm này là kết nối k-nút. Nếu không, loại bỏ nó, tăng C lên một, và loại bỏ liên kết v từ liên kết ứng viên tập $\mathbb{V}$. Sau đó, quay trở lại bước 3.
- *Bước 5:* Gán lại định tuyến chỉ dành cho những yêu cầu unicast và multicast có định tuyến qua liên kết v trong tô pô tốt nhất hiện tại.
- *Bước 6:* Tính toán tổng chi phí của tất cả các liên kết. Nếu chi phí tô pô được cải thiện, chấp nhận tô pô tạm này là tô pô tốt nhất hiện tại. Sau đó, quay trở lại bước 2. Nếu không phải, loại bỏ các cấu hình tạm thời, tăng C lên một, và loại bỏ liên kết v từ liên kết ứng viên tập $\mathbb{V}$. Sau đó, quay trở lại bước 3.
- *Bước 7:* Thoát ra và trả về tô pô tốt nhất hiện tại.

Trong bước 3, giá trị $C_{\max} = [\mathbb{U} \cdot k/2]$ cho mỗi tô pô tốt nhất là để k-nút kết nối có ít nhất $[\mathbb{U} \cdot k/2]$ liên kết. Thuật toán tối ưu chi phí cục bộ trong xử lý xóa liên kết cây multicast được thể hiện trong Hình 3.6.

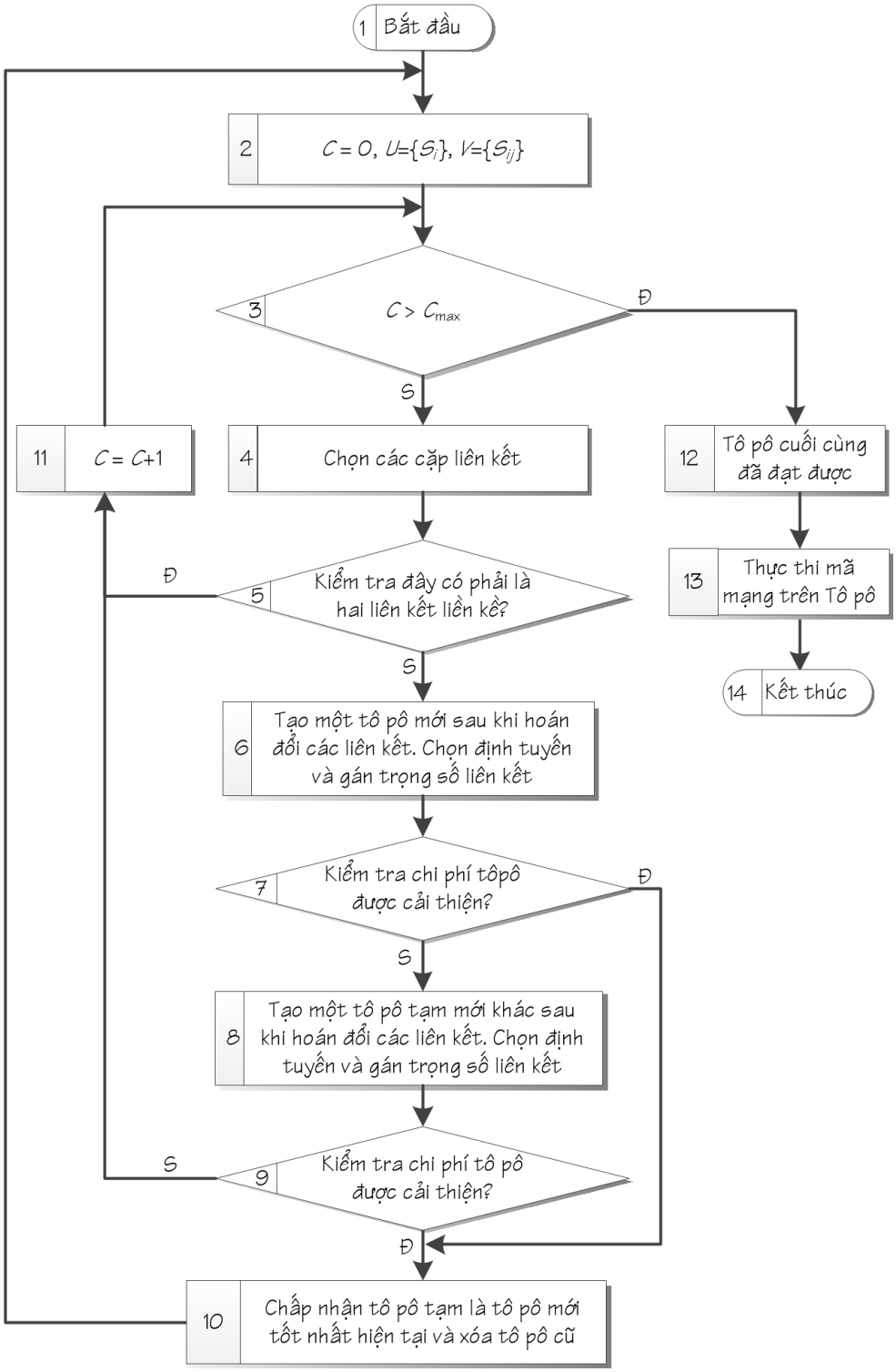
3.2.2.2. Đề xuất thuật toán thêm liên kết trong cây multicast

Luận án trình bày thuật toán thêm liên kết trong cây multicast cũng bao gồm hai pha: pha bắt đầu tạo tô pô và pha thực hiện tiến trình tối ưu cục bộ. Trong đó, pha thực hiện tiến trình tối ưu cục bộ giống pha thực hiện tiến trình tối ưu cục bộ của thuật toán xóa liên kết.

Ý tưởng chính của pha bắt đầu tạo tô pô là tạo ra một cấu hình k-nút kết nối là một tô pô chi phí thấp và sau đó tiếp tục xây dựng một tô pô mới dựa trên tô pô chi phí thấp. Pha này bao gồm các bước cụ thể như sau:

- *Bước 1:* Chọn số nút i ngẫu nhiên từ 1 đến $\mathbb{U}$.
- *Bước 2:* Xác định các nút với đỉnh nhỏ nhất. Gọi nút này X. Nếu có một số nút ứng viên, chọn một trong các chỉ số nhỏ nhất. Xác định các nút với đỉnh nhỏ nhất chưa kết nối với X. Gọi nút này Y. Nếu có một số nút ứng viên, chọn một trong số đó gần nhất đến X. Thêm liên kết X, Y.
- *Bước 3:* Lặp lại bước 2 cho đến khi liên kết của mỗi nút là nhỏ hơn hoặc bằng k.
- *Bước 4:* Kiểm tra xem cấu hình hiện tại là k-nút kết nối. Nếu đúng, đi đến bước 6.
- *Bước 5:* Kiểm tra xem các kết nối của cấu hình hiện tại có thể được tăng lên bằng cách chỉ thêm một liên kết. Nếu được, bổ sung các liên kết ngắn nhất mà bổ sung này có thể tăng khả năng kết nối. Nếu không, loại bỏ các cấu hình hiện tại và quay trở lại bước 1. Lặp lại các hoạt động nêu trên cho đến khi cấu hình hiện tại là k-nút kết nối hoặc cho đến khi kết nối của cấu hình hiện tại không thể tăng liên kết nào nữa.
- *Bước 6:* Chọn định tuyến cho từng yêu cầu và xác định trọng số liên kết.
- *Bước 7:* Thoát ra và trả về tô pô tốt nhất hiện tại.

Ở bước 5, nếu có một liên kết phải thêm vào liên kết khác để tăng kết nối. Các quy tắc là khá phức tạp để xác định liên kết có thích hợp được bổ sung vào để đảm bảo rằng các tô pô kết quả có một chi phí thấp.

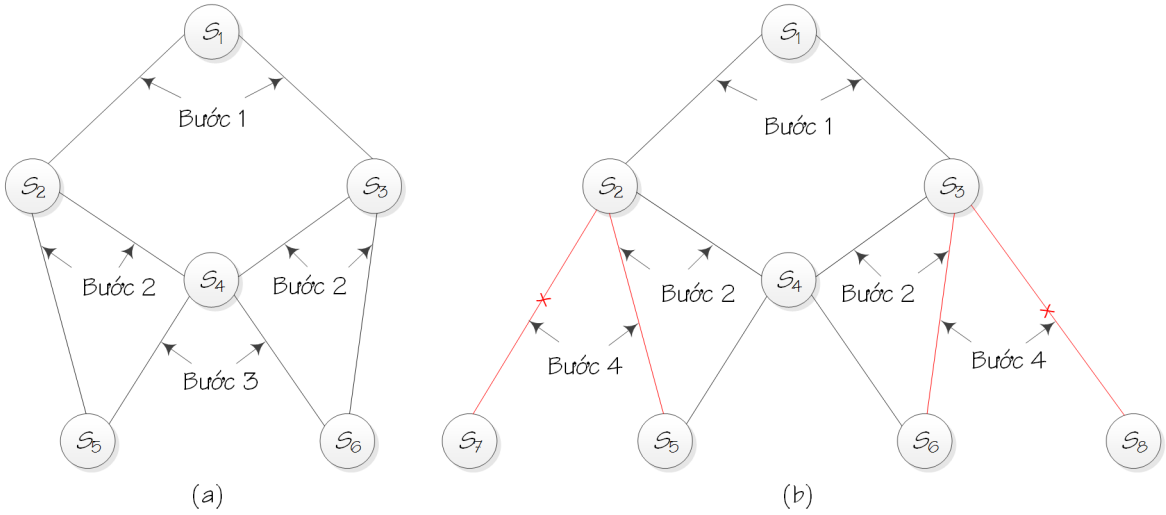


Hình 3.6: Thuật toán tối ưu cục bộ

Chi phí để đạt được tô pô tốt hơn nếu kỹ thuật mã mạng được sử dụng hỗ trợ truyền multicast, Luận án nghiên cứu sự khác biệt chi phí tô pô giữa ba trường hợp sau:

1. Mỗi yêu cầu multicast được coi là đa yêu cầu unicast.
2. Các yêu cầu multicast được xem xét riêng rẽ từ yêu cầu unicast.
3. Thuật toán cây Steiner được sử dụng để xây dựng định tuyến multicast.

Dựa vào Hình 3.4 để thực hiện hai thuật toán thêm liên kết và xóa liên kết để rút gọn cây multicast. Theo Hình 3.7.a, các bước thực hiện thuật toán thêm liên kết đạt được tô pô cuối cùng. Tuy nhiên, khi so sánh với kết quả của thuật toán xóa liên kết Hình 3.7.b ta nhận thấy rằng tập đích sẽ nhận được thông tin truyền có thể bị trùng lặp bởi có hai kết nối đến chúng. Nhưng kết quả trong Hình 3.7.a vẫn đảm bảo được các bước khi thực hiện thuật toán và bảo toàn k-nút kết nối. Vì vậy, trong quá trình thực hiện ta có thể kết hợp cả hai thuật toán để rút gọn cây multicast.



Hình 3.7: Kết quả thuật toán thêm liên kết (a) và xóa liên kết (b)

Thuật toán truyền multicast mã mạng cơ sở với chi phí cực tiểu được sử dụng để xây dựng định tuyến multicast. Để đạt được luồng cực đại trong tô pô, Luận án kết hợp rút gọn cây multicast với *song song hóa thuật toán Ford Fulkerson*.

3.2.2.3. Đề xuất song song hóa thuật toán Ford Fulkerson

Thuật toán Ford Fulkerson là thuật toán cho phép tính toán luồng cực đại trong một mạng truyền thông được đề xuất bởi tác giả Stone trong [118]. Giải pháp song

song hóa của Luận án tập trung vào tìm lưu lượng cực đại từ nguồn đến tập đích với k-nút kết nối theo phương thức truyền multicast. Theo Mục 1.1.3, mục tiêu của Luận án là xây dựng Thuật toán xử lý song song các đường dẫn tồn tại từ nguồn đến đích với điều kiện tất cả các cạnh trên đường dẫn đó vẫn còn khả năng thông qua. Xét đồ thị theo Hình 1.25, thuật toán cải tiến sẽ tìm lưu lượng gửi đi dọc theo các đường đi đó, sau đó tiếp tục tìm các đường đi khác thông qua lặp lại các bước thuật toán. Dường dẫn còn khả năng thông qua là đường dẫn có khả năng mở rộng thêm hay đường dẫn mà lưu lượng qua đó còn khả năng tăng thêm. Thuật toán kết thúc khi đạt được đường dẫn với lưu lượng có giá trị lớn nhất.

Thuật toán Ford-Fulkerson có hai bước chính: Bước thứ nhất là một tiến trình ghi nhãn để tìm kiếm cho một đường dẫn tăng lưu lượng, tức là, một đường đi từ S_0 đến $S_{\mathbb{U}}$ với ràng buộc $ll < ts$ dọc theo tất cả các các cạnh về phía trước và $ll > 0$ dọc theo tất cả các cạnh ngược lại. Nếu bước thứ nhất tìm thấy một đường dẫn có khả năng tăng lưu lượng, bước thứ hai thay đổi lưu lượng phù hợp. Nếu không còn đường dẫn tăng lưu lượng tồn tại thì thuật taoán trả về lưu lượng tối đa.

Thuật toán bắt đầu với bất kỳ lưu lượng khả thi nào ($ll = 0$), một nút ở một trong ba trạng thái: không gắn nhãn, gắn nhãn và đã duyệt, hoặc gắn nhãn và chưa duyệt. Khi thực hiện bước 1, tất cả các nút không gắn nhãn. Bước 1 chỉ ra rằng nút nguồn được gắn nhãn và chưa duyệt. Các bước chi tiết thể hiện qua Thuật toán 5.

Yêu cầu ràng buộc đối với bài toán luồng cực đại là tìm giá trị cực đại:

- Ràng buộc trọng số: $0 \leq ll^p_{S_{ij}} \leq ts_{S_{ij}}$.
- Đối xứng lưu lượng ($\forall S_i \neq S_0, S_{\mathbb{U}}$, lưu lượng dòng vào bằng với lưu lượng dòng ra): $ll^p_{S_{ij}} = -ll^p_{S_{ji}}$.
- Ràng buộc lưu lượng ($\forall S_i \neq S_0, S_{\mathbb{U}}$): $\sum_{j=1}^{\mathbb{U}} ll_{S_{ij}} = 0$.

Song song hóa thuật toán Ford Fulkerson của Luận án tập trung vào giải pháp truyền thông nhóm trong thuật toán thể hiện trong bước 2. Giải pháp này xử lý tất cả các nút được gắn nhãn và chưa duyệt với tất cả k-nút kết nối. Cải tiến bước 2 thể hiện như sau: Đối với mỗi cạnh $S_{jk} \in \mathbb{V}$ một trong hai tình huống có thể có thể xảy ra:

- S_j đã được gắn nhãn và chưa duyệt, S_i không gắn nhãn và $ll_{S_{ji}} < ts_{S_{ji}}$. Nếu tình huống này xuất hiện thì thực hiện gắn nhãn $(S_j; +; nhan\,(S_i))$ cho nút S_i, trong

Thuật toán 5: *Song song hóa thuật toán Ford Fulkerson*

Input : Tập các nút, cạnh, lưu lượng $(\mathbb{U}, \mathbb{V}, ll)$

Output: Tập Max-Flow Min-Cut

Khởi tạo, gắn nhãn nguồn $(S_0; nhan\,(S_0) = \infty)$;

Chọn bất kỳ nút S_i, có gắn nhãn và chưa duyệt (Nếu không có các nút được gắn nhãn và chưa duyệt, thì lưu lượng hiện tại là lưu lượng tối đa).;

foreach $S_{ij} \in \mathbb{V}$ **do**

  if S_j *không gắn nhãn* **then**

    if $S_{ij} \in \mathbb{V}$ *và* $ll_{S_{ij}} < ts_{S_{ij}}$ **then**

      gắn nhãn $(S_i; +; nhan\,(S_j))$ cho nút S_j:

      $nhan(S_j) = \min(nhan(S_i), ts_{S_{ij}} - ll_{S_{ij}})$;

      S_j được gắn nhãn và đã duyệt;

      S_i được gắn nhãn và chưa duyệt;

      $S_t = S_j$;

    if $S_{ji} \in \mathbb{V}$ *và* $ll_{S_{ij}} > 0$ **then**

      gắn nhãn $(S_i; -; nhan\,(S_j))$ cho nút S_j:

      $nhan(S_j) = \min(nhan(S_i), ll_{S_{ji}})$;

      S_i được gắn nhãn và đã duyệt;

      S_j được gắn nhãn và chưa duyệt;

      $S_t = S_i$;

  while $S_t \neq S_0$ **do**

    if $S_t := (S_y; +; nhan\,(S_t))$ **then**

      $ll_{S_{yt}} = ll_{S_{yt}} + nhan(S_t)$;

    if $S_t{:=}(S_y; -; nhan\,(S_t))$ **then**

      $ll_{S_{ty}} = ll_{S_{ty}} - nhan(S_t)$;

    $S_t = S_y$;

  trở lại dòng 2;

return *Max-Flow Min-Cut*$(S_0, S_{|\mathbb{U}|})$;

đó: $nhan(S_i) = \min(nhan(S_j), ts_{S_{ji}} - ll_{S_{ji}})$. Thực hiện đặt S_j đã được gắn nhãn và đã duyệt, S_i được gắn nhãn và chưa duyệt, gán nút S_j cho nút trung gian S_t;

- S_i được gắn nhãn và chưa duyệt, S_j không gắn nhãn và $ll_{S_{ji}} > 0$. Nếu tình huống này xuất hiện thì thực hiện gắn nhãn $(S_i; -; nhan\,(S_j))$ cho nút S_j, trong đó: $nhan(S_j) = \min(nhan(S_i), ll_{S_{ji}})$. Thực hiện đặt S_i đã được gắn nhãn và đã duyệt và S_j đã được gắn nhãn và chưa duyệt, gán nút S_i cho nút trung gian S_t;

Nếu nút S_t được gắn nhãn thì đi đến bước 3, ngược lại trở lại bước 2. Các bước thực thi thể hiện qua sơ đồ thuật toán Hình 3.8.

Dựa vào tô pô đã tạo Mục 1.3.3.1 trong Chương 1, song song hóa thuật toán Ford Fulkerson tiến hành gán nhãn và xây dựng tập lát cắt cực tiểu từ nút nguồn đến các nút trung gian và nút đích trong tô pô. Thuật toán sẽ tính toán luồng cực đại từ nguồn đến đích thông qua định tuyến giữa các nút. Kết quả tính giá trị luồng cực đại từ nút nguồn đến các nút đích trong tô pô đã tạo thể hiện ở Bảng 3.1

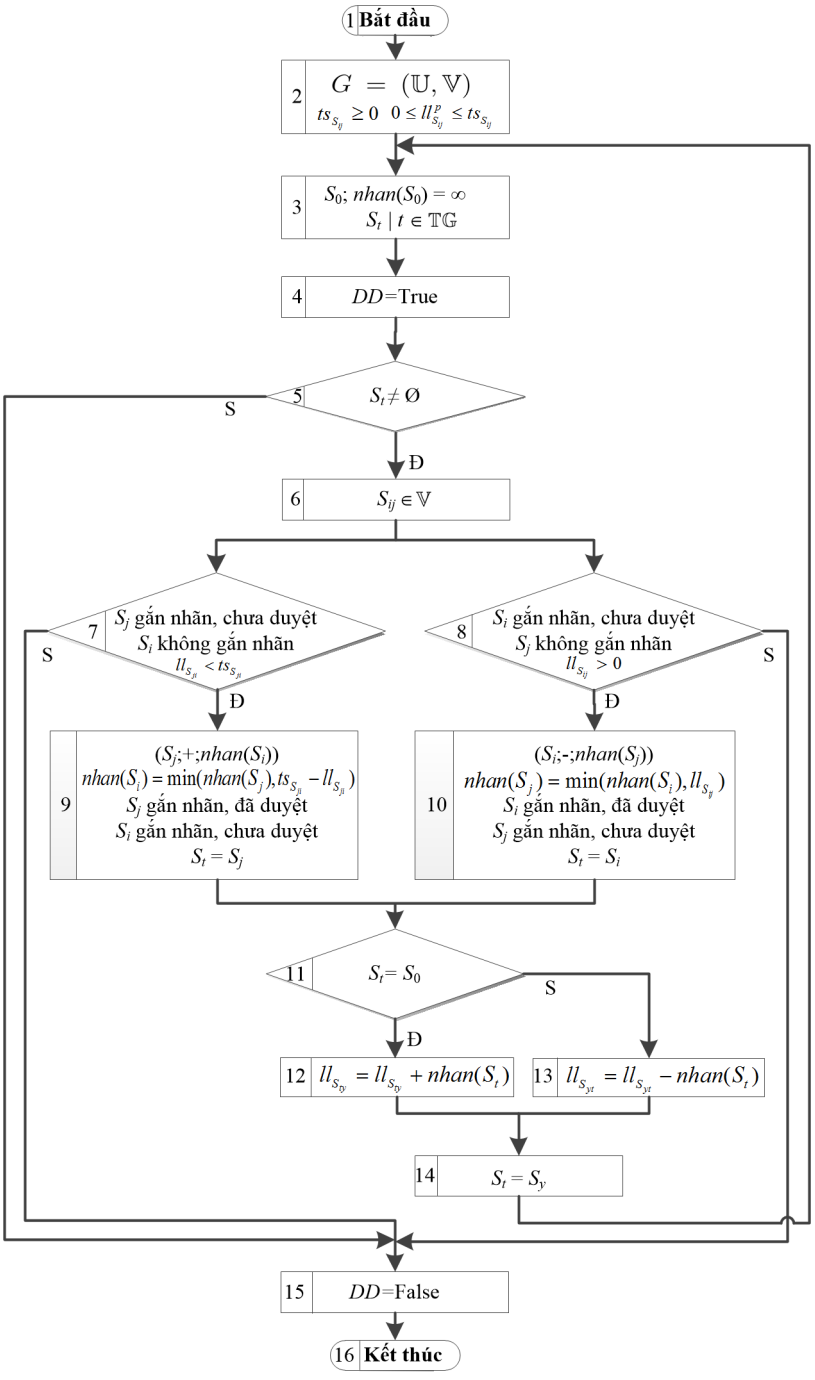
Bảng 3.1: Kết quả giá trị dựa trên song song hóa thuật toán Ford Fulkerson

Nút	1	2	3	4	5	6	7	9	10	11	12	15	18
Giá trị	27.44	27.44	27.44	13.57	10.30	23.29	27.44	24.05	13.13	24.25	27.44	14.14	27.44

Nút	20	25	27	30	39	47	48	51	54	61	71	76	78
Giá trị	27.44	10.30	10.26	10.30	22.56	12.05	10.30	10.15	27.12	13.13	27.44	10.30	17.14

Song song hóa thuật toán Ford Fulkerson để tính giá trị luồng cực đại, tuy nhiên định tuyến của thông điệp chưa phải là đường đi ngắn nhất. Thuật toán thể hiện giá trị tính toán luồng cực đại từ nguồn đến đích qua Hình 3.9.a. Để nhận xét một cách trực quan, Hình 3.9.b biểu diễn kết quả Hình 3.9.a thể hiện thông qua đồ thị, thuật toán xử lý với k-nút kết nối là 2 thì tại nút nguồn 0 có 2 định tuyến xuất phát để tính lưu lượng cực đại. Đối với thuật toán kết hợp với mã mạng, yêu cầu cần thiết là tại các nút đích phải có nút trung gian gần nhất để thực hiện phép tính XOR để tránh trùng lặp thông tin khi đến đích.

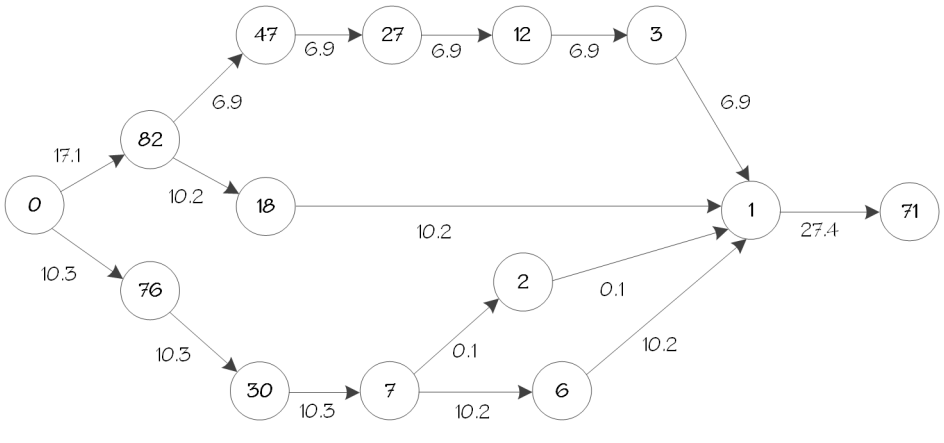
Trong giải pháp xử lý đồ thị tuyến tính cho các ứng dụng phân tán sử dụng phương thức truyền multicast thì đường dẫn từ nguồn đến đích có thể tồn tại nhiều đường đi



Hình 3.8: Sơ đồ song song hóa thuật toán Ford Fulkerson

```
Max flow from 0 to 71
   0->82 17.137780840129547/17.137780840129547
   0->76 10.300250523302278/10.300250523302278
   1->71 27.438031363431826/103.7707730144837
   2->1 0.13807224556338937/173.3547070775228
   3->1 6.940867236199129/22.399929844294924
   6->1 10.162178277738889/27.268578834024947
   7->6 10.162178277738889/10.162178277738889
   7->2 0.13807224556338937/13.405771039819596
   12->3 6.940867236199129/40.173380307117824
   18->1 10.196913603930419/10.196913603930419
   27->12 6.940867236199129/41.17808718558616
   30->7 10.300250523302278/10.33726885201113
   47->27 6.940867236199129/10.262745710257473
   76->30 10.300250523302278/29.347007500104446
   82->47 6.940867236199129/12.054926436918477
   82->18 10.196913603930419/11.9729766215762
Max flow value = 27.438031363431826
```

(a)



(b)

Hình 3.9: Tính toán luồng cực đại từ nguồn đến đích

khác nhau và phụ thuộc vào k-nút nết nối. Đối với bài toán k-nút kết nối là 2 theo Hình 3.9 thì từ nguồn có 2 đường dẫn đến đích theo phân chia tỷ lệ lưu lượng, do đó song song hóa thuật toán Ford Fulkerson sẽ thực hiện theo các ràng buộc và tham số khác nhau trong nghiên cứu của Luận án.

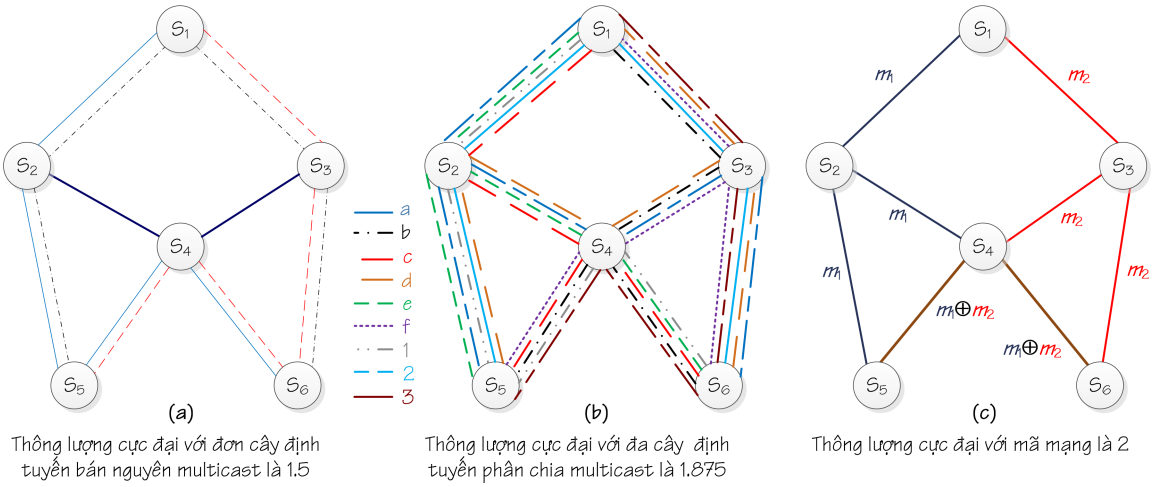
3.2.3. Xác định tỷ lệ lưu lượng trong cây multicast với mã mạng

Sau khi thực thi tô pô ban đầu của mạng $G = (\mathbb{U}, \mathbb{V})$ để đạt được một tô pô tối ưu, chi phí khi thực hiện truyền trong tô pô với trường hợp có mã mạng và không có mã mạng thể hiện qua Công thức (3.28). Theo kết quả Công thức (3.21), $1/2kn(\mathbb{U}) \leq g(\mathbb{U})$ và $thl(\mathbb{U}) \leq kn(\mathbb{U})$ có nghĩa là định tuyến phân chia được cho phép, do đó $1/2thl(\mathbb{U}) \leq g(\mathbb{U})$ là ưu điểm mã mạng, như vậy ta có tỷ lệ lưu lượng $thl(\mathbb{U})/g(\mathbb{U}) \leq 2$. Các liên kết được gán nhãn với cùng ký tự trong một cây Steiner, trong Hình 3.10.a cây gắn nhãn với 'a' trong cạnh nét liền và nét chấm gạch; trong Hình 3.10.b, nhãn 'a' được gán cho nét liền và 'b' gán cho nét chấm gạch. Theo Hình 3.10, kết quả thực thi thông lượng không có mã mạng đạt được tương ứng là 1 đối với cây multicast đơn, 1.875 đối với đa cây multicast, bằng cách truyền một lưu lượng dọc theo mỗi cây Steiner với tỷ lệ lưu lượng tương đương với trọng số cây. Sau khi thông lượng tối ưu với mã mạng theo điều kiện lý tưởng là 2. Dựa vào kết quả trên, tỷ lệ giữa mã mạng và đa cây multicast là 1.067. *Hiệu năng của mã mạng* được xác định là hiệu quả truyền dựa trên tỷ lệ gói tin đến tập đích. Nếu tỷ lệ cao hơn so với các phương thức truyền khác thì hiệu năng của mã mạng được xem là hiệu năng cao. Như ví dụ đưa ra trong Hình 3.10, hiệu năng cao của mã mạng đạt được khi thực hiện phép toán XOR tại tập đích, tỷ lệ thông lượng là 2 so với định tuyến phân chia multicast.

Trên cơ sở đề xuất hai thuật toán thêm liên kết và xóa liên kết để rút gọn cây multicast. Cây multicast rút gọn kết hợp với song song hóa thuật toán Ford Fulkerson để tìm luồng cực đại. Các thuật toán trên kết hợp giữa điều khiển lệ nguồn và luồng cực đại là điều kiện thuận lợi trong tối ưu truyền thông giữa nguồn và tập đích. Dựa trên kết quả này, Luận án trình bày thực thi mã mạng trong hệ phân tán triển khai trong máy ảo đạt được hiệu năng cao trong quá trình truyền thông.

3.3. Đề xuất thuật toán xác định mã mạng tuyến tính

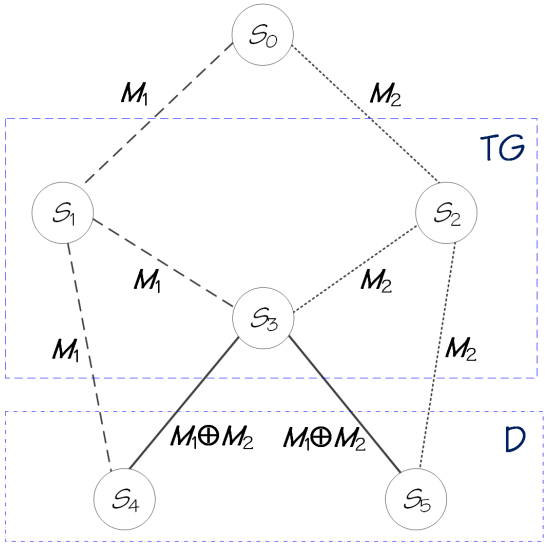
Thuật toán xác định mã mạng tuyến tính được đề xuất để giải quyết các hạn chế trình bày trong ví dụ đầu tiên trong Mục 1.3.1 của Luận án. Thuật toán xử lý trên hệ thống mạng được xét như đồ thị G. Để xét độ phức tạp của giải pháp kỹ thuật mã mạng, Luận án xét trên mô hình tổng quát và tính toán cho từng đối tượng: mã hóa



Hình 3.10: Ưu điểm của mã mạng trong cải tiến thông lượng multicast từ nguồn đến tập đích

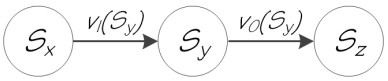
tại nút nguồn và các nút trung gian, giải mã tại tập đích. Để mô tả một cách tổng quát, đồ thị trong Hình 1.27 được trình bày lại theo Hình 3.11 để phân biệt các nút trong đồ thị. Trong đó, S_0 là nút nguồn để truyền dữ liệu đến tập đích $\mathbb{D}$ thông qua tập các nút trung gian $\mathbb{TG}$.

Đối với một nút $S_y \in \mathbb{TG}$, $v_O(S_y)$ biểu thị cung ra (id, S_y, S_z) từ S_y và $v_I(S_y)$ biểu thị cung vào (id, S_x, S_y) từ S_y theo Hình 3.12, trong đó: id là định danh của cung.



Hình 3.11: Mô hình mã mạng

Một tập trật tự $\{S_1, id_1, \ldots, S_{m-1}, id_{m-1}, S_m\}$ là đường dẫn $p(S_1, S_m)$ từ S_1 đến S_m trong G nếu (id_j, S_j, S_{j+1}), $\forall j \in \{1, \ldots, \mathbb{U}\}$. Hai đường dẫn p và p' là rời rạc nếu



Hình 3.12: Biểu thị cung vào/ra của nút S_y

chúng không chia sẻ cung chung. Một đường dẫn $p(x, z)$ là chu trình nếu $x = z$. Một đồ thị G là đồ thị không có chu trình nếu nó không chứa chu trình. Trong phần này, Luận án đưa ra độ phức tạp thời gian tổng quát của bài toán mã mạng và độ phức tạp thực thi trên các nút và cạnh trong đồ thị.

Giải pháp cơ bản của mã mạng là dữ liệu được truyền đi được chia thành nhiều gói tin để mỗi cung của mạng có thể truyền các gói tin trên mỗi kênh sử dụng. Mục đích của giải pháp là để tối đa số lượng các gói tin h chuyển từ nút nguồn S_0 đến tập đích $\mathbb{D}$, với sự ràng buộc rằng mỗi cung $v \in \mathbb{V}$ có thể được sử dụng ít nhất một lần. *Giải pháp tổng quát mã mạng của Luận án cho mạng multicast được xác định bởi mã hóa tại nút nguồn và các nút trung gian, giải mã tại tập đích.*

Các bước để xác định bài toán mã mạng dựa trên các tham số cụ thể đó là kích thước trường hữu hạn F_m, tô pô mạng hiệu quả để xây dựng mạng cực tiểu, cực tiểu chi phí và độ phức tạp mã hóa dựa trên các thuật toán mạng multicast. Kích thước của trường hữu hạn F_m được sử dụng bởi chương trình mã mạng là một tham số quan trọng xác định số tính toán được thực thi tại mỗi nút mạng.

Mạng multicast trên một đồ thị G đạt cực tiểu nếu loại bỏ bất kỳ cung v sẽ có kết quả làm giảm trọng số của nó. Một mạng cực tiểu có thể được xây dựng thông qua thuật toán bằng phép lặp loại bỏ các cung dư thừa. Luận án trình bày một thuật toán hiệu quả để xây dựng mạng cực tiểu. Mã mạng cực tiểu có một số thuộc tính kết hợp cụ thể được sử dụng để thiết kế các thuật toán mã mạng hiệu quả hơn.

Thuật toán bao gồm hai giai đoạn:

- Trong giai đoạn đầu tiên, một thuật toán lưu lượng được chạy để tìm, mỗi nút trung gian $t \in \mathbb{TG}$, tập $\mathbb{P}^t$ của đường dẫn cạnh rời rạc h từ S_0 đến $S_{\mathbb{U}}$. Chỉ có các cạnh trong sự kết hợp của những lưu lượng được xem xét trong giai đoạn thứ hai của thuật toán.
- Giai đoạn thứ hai là sử dụng thuật toán tham lam viếng thăm lần lượt mỗi cạnh và thiết kế mã tuyến tính sử dụng cho các cạnh đó. Trật tự viếng thăm các cạnh được chọn để mã cho cạnh v được thiết kế sau khi mã cho tất cả các cạnh dẫn đến v. Mục đích trong việc thiết kế các mã cho v là chọn sự kết hợp tuyến tính của các đầu vào đến nút v đảm bảo rằng tất cả các nút trung gian nằm

ở hạ lưu từ v nhận được kết hợp độc lập tuyến tính h của các ký hiệu nguồn nguyên thủy $b_1, ..., b_h$. Đối với mỗi nút trung gian S_t, thuật toán duy trì một tập $\mathbb{C}_t \subset \mathbb{V}$ và một ma trận $B_t[h \times h]$. Tập $\mathbb{C}_t$ mô tả các cạnh đã xử lý gần nhất trong mỗi đường dẫn cạnh rời rạc h trong $\mathbb{P}^t$. Các cột h của B_t tương ứng với các cạnh h trong $\mathbb{C}_t$ và cột với cạnh $c \in \mathbb{C}_t$ mô tả sự kết hợp tuyến tính $b_1, ..., b_h$ đi qua cạnh c. Đó là, nếu c truyền $b_c(l) b_1 + .. + b_c(h) b_h$, thì các cột tương ứng là $[b_c(l), \ldots, b_c(h)]^{\mathbb{T}}$. Thuật toán duy trì bất biến để ma trận B_t tại mỗi bước nghịch đảo đảm bảo rằng các bản sao của $b_1, ..., b_h$ dành cho nút trung gian S_t vẫn có thể phục hồi được với mọi sự lựa chọn mã mới.

Các tính chất của thuật toán xác định lưu lượng thông tin tuyến tính của mã mạng được mô tả như sau:

Xét đa đồ thị $G = (\mathbb{U}, \mathbb{V})$ đơn vị trọng số, có hướng, phi chu trình, để h biểu thị tập cắt tối thiểu giữa nguồn s và bất kỳ nút trung gian $t \in \mathbb{TG}$. Thuật toán lưu lượng thông tin tuyến tính theo Thuật toán 6 xây dựng mã mạng multicast tuyến tính trên một trường hữu hạn F. Trong đó, thuật toán lưu lượng thông tin tuyến tính ngẫu nhiên có thời gian thực thi kỳ vọng là đa thức $O(\mathbb{V} \cdot \mathbb{TG} \cdot h^2)$. Bất kỳ kích thước trường hữu hạn $|F| \geq 2\mathbb{TG}$ có thể được sử dụng thể hiện các ký hiệu được gửi dọc theo các cạnh. Thuật toán xác định lưu lượng thông tin tuyến tính ngẫu nhiên có thời gian thực thi $O(\mathbb{V} \cdot \mathbb{TG} \cdot h(h + \mathbb{TG}))$.

Bất kỳ kích thước trường hữu hạn $|\mathrm{F}| \geq \mathbb{TG}$ có thể được sử dụng để biểu diễn các ký hiệu được gửi dọc theo các cạnh. Kết quả các mã tuyến tính từ một trong các thuật toán lưu lượng thông tin tuyến tính có các đặc tính sau:

- Nguồn nhận ký hiệu thông tin h là đầu vào của nó.
- Cạnh v cần thời gian thực thi $O(\min(|v_I(v)|, \mathbb{TG}))$ để tính toán các ký hiệu được gửi dọc một cạnh rời. Nguồn cần thời gian thực thi $O(h)$ cho mỗi cạnh.
- Mỗi nút trung gian có thể tái tạo lại tất cả các ký hiệu thông tin h trong thời gian $O(h^2)$.

Ký hiệu *start(v)* biểu thị cạnh v bắt đầu. Đối với mỗi cạnh v chúng ta định nghĩa véc tơ mã hóa cục bộ $|v_I(start(v))|$ theo Công thức (3.31) như sau:

$$\overrightarrow{m_v} = v_I(start(v)) \rightarrow \mathrm{F}^{|v_I(start(v))|} \tag{3.31}$$

Công thức (3.31) là véc tơ xác định kết hợp tuyến tính của các ký hiệu trên các cạnh trong $|v_I(start(v))|$ tạo ra các ký hiệu trên cạnh v, đó là, nếu $y(v)$ là ký hiệu

mang theo cạnh v thì $y(v)$ được tính theo Công thức (3.32) như sau:

$$y(v) = \sum_{o}^{v_I(start(v))} \overrightarrow{m_v(o)} y(o) \tag{3.32}$$

Nhiệm vụ của Luận án là để xác định các hệ số $\overrightarrow{m_v(o)}$ để tất cả các nút trung gian có thể tái cấu trúc những thông tin nguyên thủy từ các ký hiệu. Luận án trình bày h các cạnh song song $v_1, ..., v_h$ từ một số nút mới S_0' đến S_0; các cạnh mang theo các ký hiệu đầu vào cho nguồn S_0.

Mô tả hiệu quả của tất cả các véc tơ mã cục bộ trên cạnh v độc lập của đầu vào cụ thể sử dụng véc tơ mã toàn cục $\overrightarrow{b(v)} \in \mathrm{F}^h$ đặc trưng cho sự kết hợp tuyến tính các ký hiệu đầu vào tạo ra $y(v)$. Như vậy, $\overrightarrow{b(v)} = \left[0^{p-1}, 1, 0^{h-i}\right]$ là véc tơ chiều dài h với 1 ở vị trí thứ i được tính theo Công thức (3.33) như sau:

$$\overrightarrow{b(v)} = \sum_{o}^{v_I(start(v))} \overrightarrow{m_v(o)} y(o), \forall v \in \mathbb{V} \tag{3.33}$$

véc tơ $\overrightarrow{b(e)}$ được xác định bởi vì mạng là phi chu trình và có thể được xem như chương trình mã hóa tuyến tính có thể được sử dụng cho truyền multicast từ S_0 đến $\mathbb{D}$ khi và chỉ khi $\forall S_{\mathbb{U}} \in \mathbb{D}$. Việc tái tạo các thông tin nguyên thủy có thể đạt được bằng cách xử lý hệ thống tuyến tính của phương trình qua các biến h. Mã tuyến tính trộn thông tin nhận được từ các cạnh khác nhau nhưng không bị mất thông tin cần thiết miễn là có một ánh xạ song ánh giữa đầu vào và dữ liệu đạt tại nút trung gian.

Với nhiều nút trung gian, cách tiếp cận của Luận án là thêm vào nhiều lưu lượng $S_1 - S_t$, các bước thuật toán qua các nút $S_t \in \mathbb{U}$ theo trật tự tô pô. Điều này đảm bảo rằng các véc tơ mã toàn cục của tất cả các cạnh đến $\mathbb{V}$ được biết đến khi các véc tơ mã cục bộ của các cạnh đi khỏi $\mathbb{V}$ được xác định. Thuật toán tính toán các hệ số của $\overrightarrow{m_v}$ cho các cạnh trong $v \in v_O(u)$, một cạnh tại một thời điểm. Có thể có nhiều đường dẫn lưu lượng đến nút trung gian khác nhau thông qua cạnh v. Để $\mathbb{TG}(v)$ biểu thị tập các nút trung gian sử dụng v trong một số phiên tập $\mathbb{P}^t$, tập phiên $\mathbb{P}(v)$ được tính theo Công thức (3.34):

$$\mathbb{P}(v) = \{\mathbb{P}^t \leftarrow (v) : t \in \mathbb{TG}(v)\} \tag{3.34}$$

Công thức (3.34) là tập cạnh tiền xử lý của v trong đường dẫn lưu lượng tương ứng. Hệ số khác không đối với $\overrightarrow{m_v}$ chỉ được lựa chọn cho các cạnh trong $\mathbb{P}(v)$. Để đảm

Thuật toán 6: *Thuật toán xác định mã mạng tuyến tính*

Input : Đồ thị $G = (\mathbb{U}, \mathbb{V})$, nút nguồn S_0, tập đích $\mathbb{D}$

Output: Đường dẫn rời rạc h từ S_0 đến $\mathbb{D}$, tập vectơ $\overrightarrow{m_{S_{ij}}}$, tập trọng số $\mathbb{C}_t$ và các nút trung gian a, kích thước trường hữu hạn F

$h := \min_{d \in \mathbb{D}} \min \left\{ |c| : c \text{ là tập cắt } S_0 - \mathbb{D} \right\}$;

chèn nút nguồn mới S_0' vào $\mathbb{U}$;

chèn cạnh song song $h \{v_1, ..., v_h\}$ từ S_0' đến S_0 vào $\mathbb{U}$, để $\mathbb{P}^t$ biểu thị là tập đường dẫn rời rạc h từ S_0 đến S_t;

để F là kích thước trường hữu hạn ;

foreach $i \in \mathbb{U}$ **do**

  $\overrightarrow{b(S_i)} = \left[0^{i-1}, 1, 0^{h-i}\right]$;

  $c_i := c_{S_{ij}}$;

foreach $t \in \mathbb{TG}$ **do**

  $\mathbb{C}_t := \left\{ c_{S_{01}}, .., c_{S_{it}} \right\}$;

  $\mathbb{B}_t := \left\{ \overrightarrow{b(S_{01})}, .., \overrightarrow{b(S_{it})} \right\}$;

  foreach $c_t \in \mathbb{C}_{\mathbb{TG}}$ **do**

    $a_t(c_t) := \overrightarrow{b(c_t)}$;

foreach $S_i \in \mathbb{U} \backslash \left\{ S_0' \right\}$ **do**

  foreach $v_O(S_{ij})$ **do**

    $\overrightarrow{b(S_{ij})} = \sum_{p \in v_I(start(S_{ij}))} \overrightarrow{m_{S_{ij}}(p)} y(p)$;

    $\forall t \in \mathbb{TG}(S_{ij}) : (\mathbb{B}_t \backslash \{\overrightarrow{b(\mathbb{P}^t \leftarrow (S_{ij}))}\}) \cup \left\{ \overrightarrow{b(S_{ij})} \right\}$;

    foreach $t \in \mathbb{TG}$ **do**

      tính phần tử trước của các cạnh trong $\mathbb{C}_t$;

      cập nhật $\mathbb{B}_T$;

      cập nhật a_t;

      foreach $c_t \in \mathbb{C}_{\mathbb{TG}} \backslash \{\mathbb{P}^t \leftarrow (S_{ij})\}$ **do**

        cập nhật a_t';

return $(h, \{\overrightarrow{m_{S_{ij}}} : S_{ij} \mathbb{V}\}, \{(\{c_t, a_t\}) : t \in \mathbb{TG}\}, \mathrm{F})$;

bảo rằng tất cả các nút trung gian có thể tái cấu trúc các đầu vào, thuật toán của Li và cộng sự trình bày trong [25] xác nhận rằng các véc tơ mã toàn cục $\overrightarrow{b(v)}$ là độc lập tuyến tính với số mũ tập véc tơ mã toàn cục. Thuật toán có độ phức tạp $O(\mathbb{D})$ đối với tập cạnh cần phải được kiểm tra cho mỗi $v \in \mathbb{V}$.

Nếu biểu thức $\overrightarrow{b(S_{ij})} = \sum_{p \in v_I(start(S_{ij}))} \overrightarrow{m_{S_{ij}}(p)} y(p)$ *trong Thuật toán 6 thực thi lựa chọn ngẫu nhiên* $\overrightarrow{m_v}$ *với sự hỗ trợ tập các phiên* $\mathbb{P}(v)$ *trong điều kiện* $\forall t \in \mathbb{TG}(v)$: $(\mathbb{B}_t \backslash \{\overrightarrow{b(\mathbb{P}^t \leftarrow (S_{ij}))}\}) \cup \left\{\overrightarrow{b(S_{ij})}\right\}$ *độc lập tuyến tính được thỏa thì thuật toán thực thi với thời gian* $O(\mathbb{V} \cdot \mathbb{TG} \cdot h^2)$ *và trả về thông tin cho phép mã hóa với thời gian* $O(h^2)$ *tại mỗi nút trung gian.*

Sử dụng thuật toán duyệt đồ thị đơn chúng ta có thể tìm thấy đường tăng lưu lượng từ S_0 đến $\mathbb{TG}$ trong thời gian $O(\mathbb{V})$. Chúng ta áp dụng chu kỳ thông thường này thông qua các nút trung gian cho đến khi không còn đường tăng trưởng bị sót. Chúng ta có thể tìm thấy đường tăng trưởng cho tất cả các nút trung gian trong thời gian $O(\mathbb{V} \cdot \mathbb{TG} \cdot h)$.

Các thuật toán hoạt động chính xác hơn bất kỳ trường hữu hạn của kích thước $F \geq 2\mathbb{TG}$. Để thực hiện các phép toán trường hữu hạn có hiệu quả, chúng ta có thể tạo ra một bảng tra cứu thực thể F cho phần tử tiếp sau theo nghiên cứu của Imamura trình bày trong [50]. Sử dụng bảng này, bất kỳ phép toán số học trong F có thể được tính trong thời gian hằng số.

Khởi tạo $\mathbb{C}_t$, $\mathbb{B}_t$ và $a_t(c)$ chiếm thời gian $O(\mathbb{TG} \cdot h^2)$. Hai vòng lặp chính trên tất cả các cạnh để có một tổng số lần lặp $\mathbb{V}$. Tính toán $\mathbb{P}(v)$ chiếm thời gian $O(\mathbb{TG})$ nếu lưu lượng ll^t trỏ đến phần tử trước của các cạnh trong phân chia đường dẫn của ll^t.

Tìm mã véc tơ cục bộ $\overrightarrow{m_v}$ chiếm thời gian $O(\mathbb{P}(v)) = O(\mathbb{TG})$. Tính $\overrightarrow{b(v)}$ và kiểm tra độc lập tuyến tính sử dụng véc tơ $a_t(c)$ chiếm thời gian $O(\mathbb{TG} \cdot h)$. Sau khi xác suất thành công là hằng, chi phí dự kiến cho việc tìm kiếm độc lập tuyến tính$\overrightarrow{m_v}$ là $O(1) \cdot O(\mathbb{TG} \cdot h) = O(\mathbb{D} \cdot h)$. Tính $a'_t(c) \forall t$ và tất cả $c \in \mathbb{C}_t$ với thời gian $O(\mathbb{TG} \cdot h^2)$. Kết hợp tất cả các phần trên, ta có được thời gian ràng buộc là $O(\mathbb{V} \cdot \mathbb{TG} \cdot h^2)$.

Để $S_0 < |F|$. *Xét cặp* $(k_i, l_i) \in F^h \times F^h$ *với* $k_i \cdot l_i \neq 0$ *với* $1 \leq i \leq n$. *Tồn tại kết hợp tuyến tính* $\mathbb{U}$ *của* $k_1, k_2, \ldots, k_n$ *để* $u \cdot l_i \neq 0$ *với* $1 \leq i \leq n$ *để* $\mathbb{U}$ *có thể tìm thấy trong thời gian thực thi* $O(n^2 h)$.

Ta xây dựng quy nạp $u_1, u_2, ..., u_n$ với mọi i,j với $1 \leq j \leq i \leq n$, ta có $u \cdot l_i \neq 0$. Ta chọn $u_1 = k_1$. Bây giờ, để $1 \leq j \leq i \leq n-1$.

Nếu $u_i \cdot l_i \neq 0$, ta đặt $u_{i+1} := u_i$. Ngược lại, chú ý rằng $\forall \alpha \in \mathrm{F}, (au_i + k_{i+1}) \cdot l_{i+1} \neq 0$,

$\forall j, 1 \leq j \leq i$, ta có $(au_i + k_{i+1}) \cdot l_{i+1} = 0$ nếu và chỉ nếu $\alpha = \alpha_j := -(k_{i+1}) \cdot l_i/(u_i \cdot l_i)$.

Bây giờ, để cho H là một tập hợp các các thành phần trường S_0 riêng biệt. Nếu $|H| > i, H' = H \backslash \{a_j | 1 \leq j \leq i\} \neq \emptyset$, ta chọn một số xây dựng phần tử $\alpha \in H'$ và xác định $u_{i+1} := au_i + k_{i+1}$ bằng cách xây dựng, $u_{i+1} \cdot l_i \neq 0, \forall 1 \leq j \leq i+1$.

Tính toán của tích trong mất thời gian $O(h)$. Mỗi α_j có thể được tính trong thời gian $O(h)$. Khi $|H| < S_0$, H có thể được biểu diễn theo cách khởi tạo và tìm kiếm một phần tử có thể được thực hiện trong thời gian $O(S_0)$ và loại bỏ một phần tử có thể được thực hiện trong thời gian không thay đổi. Do đó, việc tìm kiếm một hệ số đơn mất thời gian $O(nh)$ và tìm $u_1, u_2, ..., u_n$ mất thời gian $O(n^2 h)$.

Phần xác định các tính chất của thuật toán xác định lưu lượng thông tin tuyến tính được xác định tương tự xác định thời gian thực thi tại mỗi nút trong gian. Chúng ta chỉ cần thay thế thời gian dự kiến cần thiết bởi chương trình ngẫu nhiên để lựa chọn bằng thời gian cần thiết kết hợp tuyến tính. Ta có tổng thời gian thực thi theo Công thức (3.35):

$$O(\mathbb{V}(\mathbb{TG} \cdot h^2 + \mathbb{TG}^2 h)) = O(\mathbb{V} \cdot \mathbb{TG} \cdot h\,(h + \mathbb{TG})) \tag{3.35}$$

Tổng thời gian thực thi của mã mạng so với truyền multicast trình bày trong Bảng 3.2:

Bảng 3.2: Thời gian thực thi của mã mạng và truyền multicast

	Multicast	**Mã mạng**
Nút	$O(h^2)$	$O(h)$
Cung	$O(h^2\mathbb{TG})$	$O(h^2\mathbb{TG})$
Mạng	$O(\mathbb{V}\mathbb{TG}^2h^4)$	$O(\mathbb{V} \cdot \mathbb{TG} \cdot h\,(h + \mathbb{TG}))$

Trong cung cấp tài nguyên truyền thông cho hệ phân tán triển khai trong đám mây dựa trên kỹ thuật mã mạng. Giải pháp đề xuất của Luận án tối ưu hơn so với truyền thông multicast trong hệ phân tán. Độ phức tạp thời gian thực thi của thuật toán rời vào một trong ba trường hợp. Trường hợp thứ nhất, độ phức tạp thời gian của mã mạng trong trường hợp tốt nhất là $O(\mathbb{V} \cdot \mathbb{TG} \cdot h\,(h + \mathbb{TG}))$. Trường hợp thứ hai, nếu mã mạng không thực thi được tại các nút trung gian thì độ phức tạp thời gian chính là độ phức tạp thời gian của truyền multicast $O(\mathbb{V}\mathbb{TG}^2h^4)$. Trường hợp thứ ba là trường hợp xấu nhất, nghĩa là không thực hiện được cả mã mạng và truyền multicast, độ phức tạp thời gian là truyền unicast. Trường hợp thứ ba rất khó xảy ra

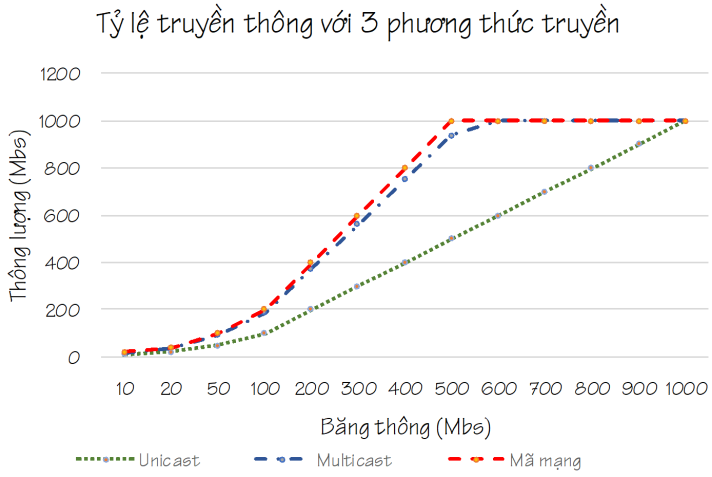
bởi khi xây dựng điều khiển cung cấp tài nguyên truyền thông với multicast dựa trên thiết lập các kênh truyền với k-nút kết nối.

Hiệu năng của kỹ thuật mã mạng trong xây dựng tô pô được trình bày bằng cách so sánh thiết kế mã mạng cơ sở với hướng unicast và đa cây multicast. Trong mô phỏng, tỷ lệ thiết lập yêu cầu unicast $tl_{S_{i,j}}(i < j)$ được chọn đồng nhất từ 10.0 đến 60.0 theo thông số đầu vào của tô pô cho các nút kết nối theo dạng token tương ứng với số nút mạng lô gíc từ 6 đến 16 theo Bảng 3.3. Trong đó, nút nguồn phân chia tỷ

Bảng 3.3: Kết quả thực thi tô pô với 3 phương thức truyền

Lưu lượng	**Mạng**		**Phương thức truyền**		
	Nút	Cạnh	Unicast	Multicast	Mã mạng
10.0	6	9	10.0	18.75	20.00
20.0	8	13	20.0	37.50	40.00
30.0	10	17	30.0	56.52	60.00
40.0	12	21	40.0	75.00	80.00
50.0	14	25	50.0	93.75	100.00
60.0	16	29	60.0	112.50	120.00

lệ truyền, các cụm nút trung gian được chọn đồng nhất về chi phí đường truyền. Dựa vào tô pô này, Luận án sẽ nghiên cứu hiệu năng của hai thuật toán được đề xuất theo phân chia tỷ lệ lưu lượng. Kết quả thể hiện ở Bảng 3.3, tập nút nguồn sau khi phân chia tỷ lệ gói truyền trên tô pô với số nút và cạnh tương ứng trong mạng con của G, các phương thức truyền được đánh giá lần lượt là multicast đơn (Unicast), đa cây multicast (Multicast) và mã mạng.



Hình 3.13: Tỷ lệ truyền thông với 3 phương thức truyền

Qua Bảng 3.3 và Hình 3.13 cho thấy, khi số lượng nút và cạnh tăng lên tương ứng với lưu lượng mạng, các tỷ lệ giữa ba phương thức truyền không có sự thay đổi lớn.

Tập nút đích nhận lưu lượng truyền với mã mạng đạt được cao hơn so với hai phương thức truyền còn lại, tỷ lệ giữa Multicast và Unicast xấp xỉ 1.875, tỷ lệ giữa Mã mạng và Multicast xấp xỉ 1.067. Vì vậy, với kết nối đa cây multicast nhất định, định tuyến chi phí cực tiểu mã mạng cơ sở có thông lượng tại tập đích cực đại. Trong kết quả mô phỏng Hình 3.13, băng thông tại tập đích có giá trị tối đa là 1000Mbs thì trường hợp k-nút kết nối là 2, thông lượng cực đại của mã mạng đạt được khi sử dụng băng thông 500Mbs trong quá trình truyền.

3.4. Đánh giá và nhận xét giải pháp kỹ thuật mã mạng cung cấp tài nguyên truyền thông cho hệ phân tán triển khai trong hệ thống ảo hóa

Dựa trên cơ sở nghiên cứu lý thuyết và các kết quả nghiên cứu liên quan, Luận án trình bày các giải thuật và thuật toán giải quyết bài toán tổng quát giải pháp điều khiển cung cấp tài nguyên cho hệ phân tán trong máy ảo dựa trên kỹ thuật mã mạng. Giải pháp này đã giải quyết vấn đề điều khiển thông điệp di chuyển trong hệ thống và các giải pháp tối ưu truyền thông điệp đối với hệ thống lớn, đa dạng và phức tạp trong môi trường Điện toán đám mây. Luận án trình bày giải pháp điều khiển cung cấp tài nguyên đối với người sử dụng và tài nguyên truyền thông trong hệ phân tán đảm bảo tính nhất quán và gắn bó. Bên cạnh đó, Luận án cũng đã trình bày giải pháp tối ưu truyền thông cho hệ thống máy chủ ảo như song song hóa thuật toán Lamport, song song hóa thuật toán Ford-Fukerson, kỹ thuật mã mạng.

Đối với giải pháp đảm bảo gắn bó trong hệ phân tán, Luận án đã song song hóa thuật toán Lamport nhằm cho phép tiến trình đi vào miền găng trong loại trừ tương hỗ phân tán sớm hơn và yêu cầu thông điệp ít hơn so với các giải pháp tương tự. Kết quả song song hóa thuật toán Lamport là một phần trong thuật toán 4PCoDT đảm bảo tính gắn bó trong hệ phân tán.

Đối với giải pháp điều khiển tỷ lệ nguồn với mã mạng, Luận án đã xây dựng được giải pháp xác định tỷ lệ nguồn và điều khiển tỷ lệ truyền truyến tính trên đồ thị con thông qua các ràng buộc của kỹ thuật mã mạng. Sau khi thuật toán xác định được tỷ lệ truyền, gói tin nguyên thủy được chia thành nhiều khối và mã hóa trước khi truyền trên các kênh. Tại tập đích, các gói tin được giải mã và trả về gói tin nguyên thủy.

Đối với giải pháp tối ưu truyền thông multicast với mã mạng, Luận án xây dựng hai thuật toán thêm liên kết và xóa liên kết nằm rút gọn cây multicast. Cây multicast sau khi được rút gọn kết hợp với thuật toán Ford Fulkerson nhằm tìm dòng tối đa truyền trên cây. Luận án đề xuất thuật toán xác định mã mạng tuyến tính nhằm đưa

ra độ phức tạp vè tổng thời gian thực thi của mã mạng.

Các nhóm giải pháp trên được thực thi trong mô phỏng hệ thống ảo hóa để tăng tính linh hoạt và khả năng điều chỉnh tối ưu thuật toán.

Tuy nhiên, những hạn chế vẫn còn tồn tại và chưa giải quyết của Luận án:

- Hệ thống ảo hóa hoạt động trên nguyên lý lô gíc, do đó hiệu năng chưa bằng so với hệ thống vật lý. Các quá trình cung cấp tài nguyên trên hệ thống ảo hóa tiến hành các tính toán, xử lý vấn còn phụ thuộc và tài nguyên mà hệ thống vật lý cung cấp.
- Giải pháp tổng quát theo Hình 1.23 chỉ áp dụng cho mô hình bài toán điều khiển cung cấp tài nguyên truyền thông được đề xuất và đưa ra độ phức tạp đối với kỹ thuật mã mạng, nếu áp dụng cho các trường hợp khác thuật toán sẽ không mang lại tính tối ưu và độ phức tạp thực thi có thể lớn hơn.

Tiểu kết Chương 3

Chương 3 đã trình bày mô hình điều khiển tỷ lệ cho multicast với mạng mã cho mạng với đồ thị con được mã. Giải pháp được phát triển trên thuật toán điều khiển thích nghi tỷ lệ tương ứng. Với mã mạng ngẫu nhiên, thuật toán được thực hiện ở tầng vận chuyển và tầng mạng để điều khiển tỷ lệ nguồn và thực thi mã mạng. Giải pháp điều khiển tỷ lệ nguồn được công bố trong công trình số (2). Ngoài ra, làm thế nào để có đồ thị con được mã tối ưu dựa trên các tiêu chí chi phí tổng quát là một vấn đề quan tâm nghiên cứu, bên cạnh đó việc rút gọn cây multicast.

Bên cạnh đó, để tối ưu hóa đồ thị con trong nghiên cứu của Luận án đã đưa ra giải pháp xây dựng tô pô truyền multicast kết hợp với mã mạng. Hai thuật toán tạo tô pô được đề xuất là thêm liên kết, xóa liên kết và thực thi chúng trong cùng ngữ cảnh để tạo các tô pô con từ tập kết nối ban đầu dựa trên các ràng buộc về thông lượng và yêu cầu xây dựng tô pô. Hai thuật toán thêm liên kết, xóa liên kết được công bố trong công trình số (4). Song song hóa thuật toán Ford Fulkerson tìm luồng cực đại từ nguồn đến tập đích theo hướng xử lý song song theo ràng buộc bảo toàn k-nút kết nối. Thuật toán triển khai truyền multicast kết hợp kỹ thuật mã mạng đảm bảo tập đích nhận thông tin không bị trùng lặp và đạt được thông lượng cực đại được công bố trong công trình số (6).

Để đạt được tính tối ưu trong truyền thông, Luận án đã trình bày mô hình điều khiển tỷ lệ cho multicast với mạng mã cho mạng với đồ thị con được mã thông qua giải pháp xây dựng tô pô truyền multicast kết hợp với mã mạng. Trong Chương 3,

Luận án trình bày độ phức tạp của bài toán mã mạng thông qua độ phức tạp mã và giải mã nút nguồn, các nút trung gian và tập đích. Bên cạnh đó, Luận án đưa ra độ phức tạp thực thi cho thuật toán mã mạng dựa trên thời gian đa thức được công bố trong công trình số (6).

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA LUẬN ÁN

Luận án đã tìm hiểu, phân tích các nghiên cứu, mô hình liên quan đến cung cấp tài nguyên trong hệ phân tán và hệ thống máy ảo. Dựa vào đó, Luận án đã đề xuất mô hình tổng quát trong cung cấp tài nguyên cho hệ phân tán trong máy ảo dựa trên kỹ thuật mã mạng. Kết quả nghiên cứu của Luận án là giải quyết mô hình tổng quát dựa trên các giải thuật, thuật toán nhằm tối ưu về điều khiển cung cấp tài nguyên dùng chung và tài nguyên truyền thông cho hệ phân tán triển khai trong máy ảo. Các kết quả cụ thể của Luận án đạt được thể hiện như sau:

- **Đối với hệ phân tán**: Luận án đã đưa ra giải pháp đảm bảo tính gắn bó, nhất quán bên trong hệ và tính trong suốt đối với người sử dụng thông qua các đóng góp:

    1. Đề xuất xử lý song song thuật toán Lamport nhằm đảm bảo trật tự tổng quát các tiến trình trong hệ phân tán thể hiện ở Chương 2. Áp dụng song song hóa thuật toán Lamport trong loại trừ tương hỗ để đảm bảo tiến trình duy nhất vào miền găng. Song song hóa thuật toán Lamport cho phép thiết lập một trật tự tổng quát chặt chẽ và ghi dấu các sự kiện diễn ra trên các máy chủ. Thuật toán cải tiến gán dấu cho sự kiện yêu cầu $3(N - 1)$ thông điệp. Khi áp dụng song song hóa thuật toán Lamport trong thuật toán loại trừ tương hỗ, tiến trình đi vào miền găng yêu cầu $(N - 1)$ thông điệp. Do đó, giải pháp cải tiến của Luận án đạt hiệu năng cao trong cải tiến thuật toán loại trừ tương hỗ phân tán.
    2. Đề xuất thuật toán 4PCoDT trình bày trong Chương 2 đảm bảo tính gắn bó dựa trên giao dịch bốn pha đạt được mục đích gắn bó mạnh trong hệ phân tán và dữ liệu nhất quán trên tất cả các bản sao. Thuật toán 4PCoDT đảm bảo tính gắn bó tập trung vào việc xây dựng thông điệp, cơ chế điều khiển, giám sát và định tuyến từ nguồn đến đích. Về cấu trúc thông điệp thể hiện đặc tính mới cho hệ phân tán để xác lập cơ chế điều khiển qua lại giữa các máy chủ nhằm hợp lực cung cấp tài nguyên dùng chung. Chương trình điều khiển thông điệp được phát triển trên nguyên lý vòng tròn ảo, dựa vào đó Luận án đưa ra thuật toán 4PCoDT nhằm đảm bảo tính gắn bó cho hệ phân tán thông qua môi trường truyền thông.

- **Đối với kỹ thuật mã mạng**: Các thuật toán được đề xuất của Luận án để đạt

được giải pháp tối ưu truyền thông trong điện toán đám mây:

1. Giải pháp kỹ thuật mã mạng được phát triển trên thuật toán điều khiển thích nghi tỷ lệ tương ứng. Khi các giá trị thay đổi về tỷ lệ, thuật toán thực hiện tính toán lại để đạt giá trị tỷ lệ cao nhất có thể. Với mã mạng ngẫu nhiên, thuật toán được thực hiện ở tầng vận chuyển và tầng mạng để điều chỉnh tỷ lệ nguồn và thực thi mã mạng. Ngoài ra, làm thế nào để có đồ thị con được mã tối ưu dựa trên các tiêu chí chi phí tổng quát là một vấn đề quan tâm nghiên cứu, bên cạnh đó việc rút gọn cây multicast và chuyển bài toán thành đa nguồn, đa tỷ lệ để tối ưu hóa chi phí.
2. Chương 3 của Luận án đã đưa ra giải pháp xây dựng tô pô truyền multicast kết hợp với mã mạng. Đề xuất hai thuật toán tạo tô pô là thêm liên kết, xóa liên kết được trình bày trong Chương 3. Hai thuật toán đề xuất cho phép rút gọn tô pô với các nút chỉ nhận và truyền thông tin. Vì vậy, xác định được nút nguồn, tập các nút trung gian và tập đích.
3. Đề xuất xử lý song song thuật toán Ford-Fukerson được trình bày trong Chương 3 tìm luồng cực đại trong mạng theo ràng buộc truyền multicast với k-nút kết nối là 2.
4. Trong Chương 3 đã đề xuất thuật toán xác định mã mạng tuyến tính. Thuật toán xác định mã mạng tuyến tính trình bày độ phức tạp tại nút nguồn, các nút trung gian và tập đích và chứng minh độ phức tạp của mã mạng tuyến tính dựa trên đa thức.

- **Đối với hệ thống Điện toán Đám mây**: Luận án trình bày giải pháp điều khiển cung cấp tài nguyên truyền thông trên cơ sở điều khiển thông điệp di chuyển trong tô pô mạng và kỹ thuật mã mạng để tối ưu truyền thông trên các kênh truyền. Trong mô phỏng Đám mây, các giải thuật của hệ phân tán và mã mạng được triệu gọi thực thi để đánh giá tổng quát mô hình bài toán.

Trên cơ sở những kết quả nghiên cứu đã đạt được của Luận án, một số hướng tiếp tục nghiên cứu, phát triển và mở rộng có thể đề xuất như sau:

- Đối với hệ phân tán, hai nhóm giải pháp cần giải quyết:
    1. Thứ nhất, phát triển giải pháp điều khiển tiến trình phòng, chống bế tắc trong cung cấp tài nguyên dùng chung.
    2. Thứ nhì, dựa vào bộ điều khiển cung cấp tài nguyên truyền thông dựa trên kỹ thuật mã mạng phát triển giải pháp cân bằng tải trên các máy chủ nhằm

đảm bảo các yêu cầu từ người sử dụng được phân tải đồng đều.

- Phát triển giải pháp điều khiển cung cấp tài nguyên truyền thông dựa trên kỹ thuật mã mạng tập trung vào các bài toán:

    1. Giải pháp mã mạng đa nguồn, đa tỷ lệ và cực tiểu độ trễ trong truyền thông.
    2. Xây dựng giao diện mở và các giao thức của bộ điều khiển cung cấp tài nguyên mã mạng để phục vụ cho các hệ thống ảo hóa khác nhau và nhiều loại ứng dụng phân tán khác nhau.

- Điện toán đám mây triển khai hệ phân tán và tối ưu truyền thông bởi giải pháp mã mạng. Vì vậy, hướng phát triển của điện toán đám mây tập trung vào tối ưu định tuyến và bảo mật các gói tin truyền giữa các hệ thống ảo với nhau.

DANH MỤC CÁC CÔNG TRÌNH CỦA TÁC GIẢ

1. *Nghiên cứu triển khai hệ thống giám sát trực tuyến các phương tiện cơ giới đường bộ*, Tạp chí Khoa học và Công nghệ Đại học Đà Nẵng, 3(64), p.163-169, 2013.
2. *Một giải pháp điều khiển tỷ lệ nguồn với mã mạng*, Hội thảo quốc gia lần thứ XVI: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, p.193-198, 2013.
3. *Giải pháp cung cấp tài nguyên truyền thông phân tán*, Kỷ yếu Hội nghị Quốc gia lần thứ VIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), p.239-245, 2015.
4. *Giải pháp tối ưu truyền thông multicast với mã mạng*, Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông, 15(35), p.28-38, 2016.
5. *Cung cấp tài nguyên truyền thông cho hệ phân tán trong máy ảo*, Tạp chí Khoa học và Công nghệ Đại học Đà Nẵng, 11(108), p.90-93, 2016.
6. *The optimal solution of communication resource allocation in distributed system integrated on cloud computing*, 6th EAI International Conference on Context-Aware Systems and Applications, p.226-236, 2017.

TÀI LIỆU THAM KHẢO

[1] D. Agrawal and A. E. Abbadi. An efficient and fault-tolerant solution for distributed mutual exclusion. *ACM Transactions on Computer Systems*, 9(1), pages 1–20, 1991.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.

[3] S. Akintoye and A. Bagula. Optimization of virtual resources allocation in cloud computing environment. In *2017 IEEE AFRICON*, pages 873–880, 2017.

[4] K. Birman and T. A. Joseph. Exploiting virtual synchrony in distributed systems. *SIGOPS Operating Systems Review*, 21(5), pages 123–138, 1987.

[5] S. Birrer, D. Lu, F. E. Bustamante, Y. Qiao, and P. Dinda. *FatNemo: Building a Resilient Multi-source Multicast Fat-Tree*, volume 3293, chapter 14, pages 182–196. Springer Berlin Heidelberg, 2004.

[6] G. S. Bobade. Network coding approach for secure cloud storage. *International Research Journal of Engineering and Technology*, 3(11), pages 979–984, 2016.

[7] R. Boutaba, Q. Zhang, and M. F. Zhani. *Virtual machine migration in cloud computing environments: Benefits, challenges, and approaches*. IGI-Global, 2013.

[8] R. Buyya, A. Beloglazov, and J. H. Abawajy. Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *Computing Research Repository*, 1006.0308, pages 6–20, 2010.

[9] R. Buyya, J. Broberg, and A. M. Goscinski. *Cloud Computing Principles and Paradigms*. Wiley Publishing, 2011.

[10] G. Cao and M. Singhal. On coordinated checkpointing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 9(12), pages 1213–1225, 1998.

[11] O. Carvalho and G. Roucairol. On mutual exclusion in computer networks. *Communications of the ACM*, 26, pages 146–147, 02 1983.

[12] J.-M. Chang and N. F. Maxemchuk. Reliable broadcast protocols. *ACM Transactions on Computer Systems*, 2(3), pages 251–273, 1984.

[13] L. Cheng, C.-L. Wang, and F. C. M. Lau. Pvtcp: Towards practical and effective congestion control in virtualized datacenters. In *21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, 2013.

[14] G.-M. Chiu and C.-M. Hsiao. A note on total ordering multicast using propagation trees. *IEEE Transactions on Parallel and Distributed Systems*, 9(2), pages 217–223, 1998.

[15] G. V. Chockler, I. Keidar, and R. Vitenberg. Group communication specifications: a comprehensive study. *ACM Computing Surveys*, 33(4), pages 427–469, 2001.

[16] P. A. Chou and Y. Wu. Network coding for the internet and wireless networks. *IEEE Signal Processing Magazine*, 24(5), pages 77–85, Sept 2007.

[17] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Allerton Conference on Communication, Control and Computing*, 2003.

[18] N. M. M. K. Chowdhury and R. Boutaba. Network virtualization: State of the art and research challenges. *IEEE Communications Magazine*, 47(7), pages 20–26, 2009.

[19] J. R. Correa and N. E. Stier-MosesCorrea. Wardrop equilibria. *Wiley Encyclopedia of Operations Research and Management Science*, pages 1–13, 2010.

[20] D. Cotroneo, L. D. Simone, A. K. Iannillo, A. Lanzaro, R. Natella, J. Fan, and W. Ping. Network function virtualization: Challenges and directions for reliability assurance. In *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pages 37–42, 2015.

[21] G. F. Coulouris and J. Dollimore. *Distributed systems: concepts and design.* Addison-Wesley„ 4th ed. edition, 2005.

[22] D. Crisan, R. Birke, C. Minkenberg, and M. Gusat. Lossless virtual networks. In *IEEE Conference on Computer Communications Workshops*, pages 5–6, 2013.

[23] B. Cronkite-Ratcliff, A. Bergman, S. Vargaftik, M. Ravi, N. McKeown, I. Abraham, and I. Keslassy. Virtualized congestion control, 2016.

[24] S. Deb and R. Srikant. Congestion control for fair resource allocation in networks with multicast flows. *IEEE/ACM Transactions on Networking*, 12(2), pages 274–285, 2004.

[25] X. Défago, A. Schiper, and P. Urbán. Total order broadcast and multicast algorithms: Taxonomy and survey. *ACM Computing Surveys (CSUR)*, 36(4), pages 372–421, 2004.

[26] U. Divakarlaa and G. Kumarib. An overview of cloud computing in distributed systems. *AIP Conference Proceedings*, 1324, pages 184–186, 2010.

[27] U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg. Adaptive windows for duplicate detection. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1073–1083, 2012.

[28] Elayaraja.D and J. E. Fiorenza. An efficient approach for data sharing in cloud computing using digital signature. *International Journal of Computer Techniques*, 2(2), pages 1–6, 2015.

[29] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), pages 1–16, 2007.

[30] B. Ensink and V. Adve. Coordinating adaptations in distributed systems. In *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, pages 446–455, 2004.

[31] K. Erciyes and V. Adve. *Distributed Graph Algorithms for Computer Networks.* Springer Publishing Company, Incorporated, 2013.

[32] P. D. Ezhilchelvan, R. A. Macedo, and S. K. Shrivastava. Newtop: a fault-tolerant group communication protocol. In *Proceedings of 15th International Conference on Distributed Computing Systems*, pages 296–306, 1995.

[33] M. J. Farooq and Q. Zhu. Adaptive and resilient revenue maximizing resource allocation and pricing in cloud computing environments. *CoRR*, 1707.08691, pages 1–7, 2017.

[34] N. Feamster, J. Rexford, and E. Zegura. The road to sdn: an intellectual history of programmable networks. *SIGCOMM Computer Communication Review*, 44(2), pages 87–98, 2014.

[35] F. H. Fitzek, J. Heide, M. V. Pedersen, and M. Katz. Implementation of network coding for social mobile clouds [applications corner]. *Signal Processing Magazine, IEEE*, 30(1), pages 159–164, 2013.

[36] H. Frank, R. E. Kahn, and L. Kleinrock. Computer communication network design: Experience with theory and practice. In *Spring Joint Computer Conference*, AFIPS '72 (Spring), pages 255–270, New York, NY, USA, 1972. ACM.

[37] G. G and D. Bona. A survey on cloud computing elasticity. In *5th International Conference on Utility and Cloud Computing (UCC)*, page 263–270, 2012.

[38] R. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25(1), pages 73–85, 1977.

[39] G. E. Gonçalves, P. T. Endo, T. Damasceno, A. Vitor, A. Palhares, D. Sadok, J. Kelner, B. Mel, and J. erik Mångs. Resource allocation in clouds: Concepts, tools and research challenges. *SimposioBrasileiro de Redes de Computadores e SistemasDistribuidos*, pages 197–240, 2011.

[40] J. Gray. The transaction concept: virtues and limitations. In *Proceedings of the Seventh International Conference on Very Large Data Bases - Volume 7*, pages 144–154. VLDB Endowment, 1981.

[41] S. Haddad, F. Kordon, L. Pautet, and L. Petrucci. *Distributed Systems Design and Algorithms.* ISTE Ltd, 2011.

[42] A. Haider, R. Potter, and A. Nakao. Challenges in resource allocation in network virtualization, 2009.

[43] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2), pages 90–97, 2015.

[44] M. A. Hernández and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1), pages 9–37, Jan 1998.

[45] J.-M. Hélary, N. Plouzeau, and M. Raynal. A distributed algorithm for mutual exclusion in an arbitrary network. *The Computer Journal*, 31(4), pages 289–295, 1988.

[46] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *IEEE International Symposium on Information Theory*, page 442, 2003.

[47] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10), pages 4413–4430, 2006.

[48] HP. Software-defined networking and network virtualization. Technical report, 2014.

[49] F. Hu, Q. Hao, and K. Bao. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys Tutorials*, 16(4), pages 2181–2206, 2014.

[50] K. Imamura. A method for computing addition tables in gf(p^n). *IEEE Transactions on Information Theory*, 26(3), pages 367–369, 1980.

[51] R. Jain and S. Paul. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, 51(11), pages 24–31, 2013.

[52] W. Jia and W. Zhou. *Distributed Network Systems: From Concepts to Implementations (Network Theory and Applications).* Springer-Verlag New York, Inc., 2006.

[53] S. Kafaie, Y. Chen, M. H. Ahmed, and O. A. Dobre. Network coding with link layer cooperation in wireless mesh networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 3672–3677, June 2015.

[54] A. Kershenbaum. *Telecommunications network design algorithms.* McGraw-Hill, Inc., 1993.

[55] A. R. Khan. Access control in cloud computing environment. *ARPN Journal of Engineering and Applied Sciences*, 7, pages 613–615, 2012.

[56] A. Khreishah, I. M. Khalil, and J. Wu. Distributed network coding-based opportunistic routing for multicast. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 115–124, 2012.

[57] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5), pages 782–795, 2003.

[58] X. Kong, C. Lin, Y. Jiang, W. Yan, and X. Chu. Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. *Journal of Network and Computer Applications*, 34(4), pages 1068–1077, 2011.

[59] J. Krigslund, J. Hansen, D. E. Lucani, F. H. P. Fitzek, and M. Medard. Network coded software defined networking: Design and implementation. In *Proceedings of European Wireless 2015; 21th European Wireless Conference*, pages 1–6, 2015.

[60] A. D. Kshemkalyani and M. Singhal. An optimal algorithm for generalized causal message ordering, 1996.

[61] A. D. Kshemkalyani and M. Singhal. Necessary and sufficient conditions on information for causal message ordering and their optimal implementation. *Distributed Computing*, 11(2), pages 91–111, 1998.

[62] A. D. Kshemkalyani and M. Singhal. *Distributed Computing: Principles, Algorithms, and Systems.* Cambridge University Press, 2008.

[63] M. Kumar. Commit protocols in distributed database system: A comparison. *International Journal for Innovative Research in Science Technology*, 2(12), pages 277–281, 2016.

[64] S. H. Kwok, K. Y. Chan, and Y. M. Cheung. A server-mediated peer-to-peer system. *SIGecom Exch.*, 5(3), pages 38–47, 2005.

[65] V. S. Lê. *Hệ tin học phân tán.* Nhà xuất bản Đại học Quốc gia Tp. Hồ Chí Minh, 2002.

[66] V. S. Lê. Nghiên cứu phát triển giải thuật quản lý nhiều bản sao cho các ứng dụng phức tạp. *Tạp chí khoa học và Công nghệ Đại học Đà Nẵng*, 1, pages 64–71, 2003.

[67] V. S. Lê. Research on the construction of parallel-process model on distributed environment. *Tạp chí Khoa học Đại học Quốc gia Hà Nội*, 9(2), pages 46–55, 2003.

[68] V. S. Lê. Phát triển giải pháp đảm bảo gắn bó dữ liệu trong hệ thống thông tin đào tạo trên internet/intranet. *Tạp chí khoa học và Công nghệ Các Trường Đại học Kỹ thuật Việt Nam*, 46, pages 27–31, 2004.

[69] V. S. Lê, H. V. Đặng, and M. H. Hồ. Mô hình người sản xuất - người tiêu thụ trong xây dựng giải pháp kỹ thuật giám sát các hoạt động đồng thời phân tán. *Tạp chí khoa học và Công nghệ Đại học Đà Nẵng*, 6, pages 94–102, 2009.

[70] V. S. Lê and H. H. Nguyễn. Phát triển phương pháp đảm bảo gắn bó dữ liệu trong điều kiện có sự cố của các hệ thống thông tin đăng ký trên mạng internet/intranet. *Tạp chí khoa học và Công nghệ Đại học Đà Nẵng*, 6, pages 30–35, 2004.

[71] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7), pages 558–565, 1978.

[72] M. Langberg, A. Sprintson, and J. Bruck. The encoding complexity of network coding. In *International Symposium on Information Theory*, pages 1987–1991, 2006.

[73] M. Langberg, A. Sprintson, and J. Bruck. Network coding: A computational perspective. *IEEE Transactions on Information Theory*, 55(1), pages 147–157, 2009.

[74] K. Laxmi and D. R.V.Krishnaiah. A reliable grid service infrastructure. *International Journal of Computer Engineering and Applications*, 3(3), pages 191–197, 2013.

[75] Y. C. Lee and A. Y. Zomaya. Rescheduling for reliable job completion with the support of clouds. *Future Generation Computer Systems*, 26(8), pages 1192–1199, 2010.

[76] J. Li, J. Li, Z. Liu, and C. Jia. Enabling efficient and secure data sharing in cloud computing. *Concurrency and Computation: Practice and Experience*, 26(5), pages 1052–1066, 2014.

[77] Q. Li, H. Liu, H. Ning, Y. Fu, S. Hu, and S. Yang. Onf software-defined networking: The new norm for networks. Technical report, 2012. Open Networking Foundation.

[78] S.-y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2), pages 371–381, 2003.

[79] Y. Li, J. Quan, X. Tan, and H. Li. A layered multi-tree ip multicast protocol with network coding. *INFOCOMP, The Second International Conference on Advanced Communications and Computation*, pages 171 – 175, 2012.

[80] Z. Li and B. Li. Network coding in undirected networks, 2004.

[81] Y. Liu and Y. Morgan. A network coding based cloud storage scheme. *International Journal of Internet and Distributed Systems*, 03, pages 1–8, 2018.

[82] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[83] M. Maekawa. A sqrt(n) algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer Systems*, 3(2), pages 145–159, 1985.

[84] P. J. Maenhaut, H. Moens, B. Volckaert, V. Ongenae, and F. D. Turck. Resource allocation in the cloud: From simulation to experimental validation. In *IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 701–704, 2017.

[85] M. Medard and A. Sprintson. *Network coding: fundamentals and applications*. Academic, 2010.

[86] P. M. Mell and T. Grance. Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States, 2011.

[87] J. Metzler, A. Metzler, and Associates. The 2015 guide to sdn and nfv. Technical report, 2015.

[88] M.H.Nerkar and S. V. Shinkar. Cloud computing in distributed system. *International Journal of Computer Science and Informatics*, 1(4), pages 70–74, 2012.

[89] MichelRaynal. Prime numbers as a tool to design distributed algorithms. *Information Processing Letters*, 33(1), pages 53–58, 1989.

[90] S. Mishra and P. K. Srimani. Fault-tolerant mutual exclusion algorithms. *Journal of Systems and Software*, 11(2), pages 111–129, 1990.

[91] M. Mizuno, M. L. Neilsen, and R. Rao. A token based distributed mutual exclusion algorithm based on quorum agreements. In *11th International Conference on Distributed Computing Systems*, pages 361–368, 1991.

[92] N. Mohamed and T. Michel. How to detect a failure and regenerate the token in the log(n) distributed algorithm for mutual exclusion. In *Distributed Algorithms*, pages 155–166. Springer Berlin Heidelberg, 1988.

[93] M. A. Murphy and S. Goasguen. Virtual organization clusters: Self-provisioned clouds on the grid. *Future Generation Computer Systems*, 26(8), pages 1271–1281, 2010.

[94] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1), pages 90 – 100, 2003.

[95] X. Nan, Y. He, and L. Guan. Optimal resource allocation for multimedia cloud based on queuing model, 2011.

[96] S. Naseera. A distributed ring algorithm for coordinator election in distributed systems. *ICTACT Journal on Communication Technology*, 7(3), pages 1341–1344, 2016.

[97] S. Nishio, K. F. Li, and E. G. Manning. A resilient mutual exclusion algorithm for computer networks. *IEEE Transactions on Parallel and Distributed Systems*, 1(3), pages 344–356, 1990.

[98] N. T. Oanh. Thực trạng cạn kiệt ipv4 trên thế giới. Technical report, 2011.

[99] M. T. Ozsu and P. Valduriez. *Principles of Distributed Database Systems.* Prentice Hall Press, 2011.

[100] N. K. Pandey, S. Chaudhary, and N. K. Joshi. Resource allocation strategies used in cloud computing: A critical analysis. In *2016 2nd International Conference on Communication Control and Intelligent Systems (CCIS)*, pages 213–216, 2017.

[101] T. F. J.-M. Pasquier, J. Singh, D. Eyers, and J. Bacon. Camflow: Managed data-sharing for cloud services. *CoRR*, 1506.04391, 2015.

[102] L. L. Peterson and B. S. Davie. *Computer Networks, Fifth Edition: A Systems Approach.* Morgan Kaufmann Publishers Inc., 2011.

[103] K. Raymond. A tree-based algorithm for distributed mutual exclusion. *ACM Transactions on Computer Systems*, 7(1), pages 61–77, 1989.

[104] M. Raynal. *Distributed Algorithms for Message-Passing Systems.* Springer Publishing Company, Incorporated, 2013.

[105] M. Raynal, A. Schiper, and S. Toueg. The causal ordering abstraction and a simple way to implement it. *Information Processing Letters*, 39(6), pages 343–350, 1991.

[106] G. Reese. *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud.* O'Reilly Media, Inc., 2009.

[107] G. Ricart and A. K. Agrawala. An optimal algorithm for mutual exclusion in computer networks. *Communications of the ACM*, 24(1), pages 9–17, 1981.

[108] B. A. Sanders. The information structure of distributed mutual exclusion algorithms. *ACM Transactions on Computer Systems*, 5(3), pages 284–299, 1987.

[109] S. Sarkar and L. Tassiulas. A framework for routing and congestion control for multicast information flows. *IEEE Transactions on Information Theory*, 48(10), pages 2690–2708, 2002.

[110] P. Sattari, C. Fragouli, and A. Markopoulou. Active topology inference using network coding. *Physical Communication*, 6, pages 142 – 163, 2013.

[111] M. A. Sharkh, M. Jammal, A. Shami, and A. Ouda. Resource allocation in a network-based cloud computing environment: design challenges. *Communications Magazine, IEEE*, 51(11), pages 46–52, 2013.

[112] M. A. Sharkh, A. Shami, and A. Ouda. Optimal and suboptimal resource allocation techniques in cloud computing data centers. *Journal of Cloud Computing*, 6(1), pages 1–6, 2017.

[113] L. Si-Hyeon and C. Sae-Young. Capacity of a class of multicast tree networks. *IEEE Transactions on Information Theory*, 59(6), pages 3848–3857, 2013.

[114] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts*. Wiley Publishing, 2012.

[115] M. Singhal. Deadlock detection in distributed systems. *Computer*, 22(11), pages 37–48, 1989.

[116] M. Singhal. A heuristically-aided algorithm for mutual exclusion in distributed systems. *IEEE Transactions on Computers*, 38(5), pages 651–662, 1989.

[117] V. Sontakke, P. Patil, S. Waghamare, R. Kulkarni, N. S. Patil, and M. Saravanapriya. Dynamic resource allocation strategy for cloud computing using virtual machine environment. *International Journal of Engineering Science and Computing IJESC*, 6(5), pages 4804–4806, 2016.

[118] H. S. Stone. Multiprocessor scheduling with the aid of network flow algorithms. *IEEE Transactions on Software Engineering*, SE-3(1), pages 85–93, 1977.

[119] L. S. Subhash and K. Thooyamani. Allocation of resource dynamically in cloud computing environment using virtual machines. *International Journal of Advancements in Technology*, 8(4), pages 1–3, 2017.

[120] I. Suzuki and T. Kasami. A distributed mutual exclusion algorithm. *ACM Transactions on Computer Systems*, 3(4), pages 344–349, Nov. 1985.

[121] D. Szabó, A. Csoma, P. Megyesi, A. Gulyás, and F. H. P. Fitzek. Network coding as a service. *CoRR*, 1601.03201, pages 1–11, 2016.

[122] D. Szabó, F. Németh, B. Sonkoly, A. Gulyás, and F. H. Fitzek. Towards the 5g revolution: A software defined network architecture exploiting network coding as a service. *SIGCOMM Computer Communication Review*, 45(4), pages 105–106, 2015.

[123] S. E. Tan, Z. W. Siew, Y. K. Chin, S. C. K. Lye, and K. T. K. Teo. Minimizing network coding nodes in multicast tree construction via genetic algorithm. In *2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 399–404, 2012.

[124] A. S. Tanenbaum and M. v. Steen. *Distributed Systems: Principles and Paradigms.* CreateSpace Independent Publishing Platform, 2006.

[125] V. N. T. P. M. Team. Vmware® nsx for vsphere network virtualization design guide ver 3.0. Report, 2014.

[126] G. Tel. *Introduction to distributed algorithms.* Cambridge University Press, 1994.

[127] N. H. P. Trần and V. S. Lê. Nghiên cứu mô hình giám sát trực tuyến hệ thống mạng phân tán quy mô lớn. In *Hội thảo Quốc gia lần thứ XIII về CNTT và Truyền thông*, pages 239–250, 2011.

[128] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely. Dynamic resource allocation and power management in virtualized data centers, 2010.

[129] M. G. Velazquez. A survey of distributed mutual exclusion algorithms. Technical report, 1993.

[130] T. Velte, A. Velte, and R. Elsenpeter. *Cloud Computing, A Practical Approach.* McGraw-Hill, Inc., 2010.

[131] V.Vinothina, R.Sridaran, and PadmavathiGanapathi. A survey on resource allocation strategies in cloud computing. *International Journal of Advanced Computer Science and Applications*, 3, pages 97–104, 2012.

[132] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Institution of Civil Engineers*, pages 325–362, 1952.

[133] S. E. Whang, D. Marmaros, and H. Garcia-Molina. Pay-as-you-go entity resolution. *IEEE Trans. on Knowl. and Data Eng.*, 25(5), pages 1111–1124, May 2013.

[134] J. Widmer, C. Boutremans, and J.-Y. L. Boudec. End-to-end congestion control for tcp-friendly flows with variable packet size. *Computer Communication Review*, 34(2), pages 137–151, 2004.

[135] Y. Wu, M. Chiang, and S.-Y. Kung. Distributed utility maximization for network coding based multicasting: A critical cut approach. In *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, 2006.

[136] F. Xhafa and A. Abraham. Computational models and heuristic methods for grid scheduling problems. *Future Generation Computer Systems*, 26(4), pages 608–621, 2010.

[137] Y. Xi and E. M. Yeh. Distributed algorithms for minimum cost multicast with network coding. *IEEE/ACM Transactions on Networking*, 18(2), pages 379–392, 2010.

[138] N. Yadav, S. Yadav, and S. Mandiratta. A review of various mutual exclusion algorithms in distributed environment. *International Journal of Computer Applications*, 129, pages 11–16, 2015.

[139] W. Yan, W. Jinkuan, and S. Jinghao. A game-theoretic based resource allocation strategy for cloud computing services. *Scientific Programming*, 2016, pages 1–13, 2016.

[140] R. W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *IEEE Transactions on Information Theory*, 45(4), pages 1111–1120, 1999.

[141] A. Zahid and R. Sharif. Content matching – detection of duplicate and near duplicate videos. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, 2(4), 2013.

[142] M. R. Zakerinasab and M. Wang. An update model for network coding in cloud storage systems. In *Communication, Control, and Computing*, pages 1158–1165, 2012.

[143] L. Zhang, S. Lai, C. Wu, Z. Li, and C. Guo. Virtualized network coding functions on the internet. In *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 129–139, 2017.

[144] S. Zou, X. Wen, K. Chen, S. Huang, Y. Chen, Y. Liu, Y. Xia, and C. Hu. Virtualknotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter. In *IEEE 32nd International Conference on Distributed Computing Systems*, pages 12–21, 2012.