

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM**

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ



Vũ Việt Dũng

**NGHIÊN CỨU CÔNG NGHỆ ẢO HÓA CHỨC NĂNG MẠNG (NFV)
VÀ ĐỀ XUẤT PHƯƠNG PHÁP CẢI TIẾN HIỆU NĂNG**

LUẬN VĂN THẠC SĨ MÁY TÍNH

Hà Nội – 2026

BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Vũ Việt Dũng

NGHIÊN CỨU CÔNG NGHỆ ẢO HÓA CHỨC NĂNG MẠNG (NFV)
VÀ ĐỀ XUẤT PHƯƠNG PHÁP CẢI TIẾN HIỆU NĂNG

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC:
TS. Ngô Hải Anh

Hà Nội – 2026

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả được công bố với các tác giả khác đều được sự đồng ý của các đồng tác giả trước khi đưa vào luận văn. Các kết quả nêu trong luận văn là trung thực và chưa từng được công bố trong bất kỳ luận văn hoặc công trình khoa học nào khác.

Hà Nội, ngày 06 tháng 7 năm 2026

HỌC VIÊN



Vũ Việt Dũng

LỜI CẢM ƠN

Luận văn được thực hiện tại Học viện Khoa học và Công nghệ – Viện Hàn lâm Khoa học và Công nghệ Việt Nam, dưới sự hướng dẫn của TS. Ngô Hải Anh. Tôi xin bày tỏ lòng biết ơn sâu sắc đến Thầy về định hướng khoa học, người đã động viên, trao đổi nhiều kiến thức và chỉ bảo tôi vượt qua những khó khăn để hoàn thành luận văn này. Đồng thời, tôi cũng xin chân thành cảm ơn tới các nhà khoa học, tác giả của các công trình công bố đã được trích dẫn trong luận văn, đây là những tư liệu quý, kiến thức liên quan quan trọng giúp Học viên hoàn thành luận văn. Xin cảm ơn đến các nhà khoa học đã phản biện các công trình nghiên cứu của Học viên.

Tôi trân trọng cảm ơn Viện Công nghệ thông tin – Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình nghiên cứu thực hiện luận văn.

Cuối cùng, tôi xin gửi lời cảm ơn sâu sắc tới gia đình, bạn bè, những người đã luôn ủng hộ, giúp đỡ và hỗ trợ tôi về mọi mặt để tôi yên tâm học tập đạt kết quả tốt.

Hà Nội, ngày 06 tháng 7 năm 2026

HỌC VIÊN



Vũ Việt Dũng

MỤC LỤC

Lời cam đoan	1
Lời cảm ơn	2
MỤC LỤC	i
Danh mục các bảng	
Danh mục các hình vẽ, đồ thị	
MỞ ĐẦU	1
Lý do chọn đề tài	1
Mục đích nghiên cứu	2
Nội dung nghiên cứu	2
Cơ sở khoa học và tính thực tiễn của đề tài	3
Những đóng góp của luận văn	4
Đối tượng nghiên cứu	4
Phương pháp nghiên cứu	5
1. Tổng quan về công nghệ ảo hóa chức năng mạng (NFV)	6
1.1. Đặt vấn đề	6
1.2. Tính cấp thiết của đề tài	8
1.3. Mục tiêu, đối tượng và phạm vi nghiên cứu	10
1.4. Phương pháp nghiên cứu	12
1.5. Ý nghĩa khoa học và thực tiễn của đề tài	14
1.6. Bố cục của luận văn	16
2. Nghiên cứu các phương pháp cải tiến hiệu năng NFV	18
2.1. Giới thiệu chương	18
2.2. Tổng quan về công nghệ NFV, SDN và SFC	20
2.2.1. Khái niệm và đặc trưng của NFV	20
2.2.2. Khái niệm và vai trò của SDN	21
2.2.3. Khái niệm và vai trò của SFC	22
2.2.4. Mối quan hệ giữa NFV, SDN và SFC	23

2.3. Kiến trúc và các thành phần chính của hệ thống NFV	24
2.3.1. Kiến trúc tổng quát của hệ thống NFV	25
2.3.2. Hạ tầng NFV và vai trò của NFVI	25
2.3.3. Mặt phẳng quản lý và điều phối NFV	26
2.3.4. Mặt phẳng dịch vụ và vòng đời VNF	28
2.3.5. Một số lựa chọn thiết kế nền tảng và ảnh hưởng đến hiệu năng . . .	29
2.4. Các yêu cầu kỹ thuật và chỉ tiêu hiệu năng trong môi trường NFV/SFC .	30
2.4.1. Yêu cầu về tài nguyên tính toán và tài nguyên mạng	31
2.4.2. Yêu cầu về độ trễ và chất lượng dịch vụ	31
2.4.3. Yêu cầu về thông lượng và khả năng phục vụ dịch vụ	32
2.4.4. Yêu cầu về hiệu quả sử dụng tài nguyên	32
2.4.5. Yêu cầu về khả năng mở rộng và thích ứng động	33
2.4.6. Yêu cầu về khả năng chia sẻ, cô lập và quản lý trạng thái	33
2.5. Các nhóm phương pháp cải tiến hiệu năng NFV	34
2.5.1. Nhóm phương pháp phân bổ tài nguyên và điều phối dịch vụ . . .	35
2.5.2. Nhóm phương pháp đặt và xâu chuỗi VNF	35
2.5.3. Nhóm phương pháp tối ưu hiệu năng thực thi	36
2.5.4. Nhóm phương pháp tăng độ tin cậy, khả năng phục hồi và khôi phục lỗi	37
2.5.5. Nhận xét tổng hợp về các nhóm phương pháp	37
2.6. Phân tích các hướng liên quan và khoảng trống nghiên cứu	38
2.6.1. Các hướng placement có xét độ trễ và yêu cầu SLA	38
2.6.2. Các hướng placement có xét workload động theo thời gian	39
2.6.3. Các hướng nghiên cứu về đa thuê, sharing và phụ tải tài nguyên nền	39
2.6.4. Các hướng heuristic thay cho tối ưu hóa chính xác	40
2.6.5. Khoảng trống nghiên cứu và định hướng của luận văn	40
2.7. Kết luận chương	41
3. Đề xuất phương pháp cải tiến hiệu năng cho môi trường NFV	43
3.1. Giới thiệu chương	43
3.2. Cơ sở đề xuất phương pháp và bài toán đặt ra	44
3.3. Mô hình hệ thống và giả thiết mô phỏng	45
3.4. Thiết kế phương pháp DWSA-VNF	47
3.4.1. Độ phức tạp tính toán và khả năng mở rộng	53
3.5. Môi trường cài đặt và mô phỏng	54

3.6. Kết luận chương	55
4. Mô phỏng và đánh giá kết quả	57
4.1. Giới thiệu chương	57
4.2. Thiết lập môi trường mô phỏng và kịch bản thí nghiệm	58
4.3. Các chỉ số đánh giá	61
4.4. Kết quả so sánh theo khả năng phục vụ và hiệu năng dịch vụ	62
4.5. Kết quả so sánh theo hiệu quả sử dụng tài nguyên và chi phí vận hành . .	66
4.6. Kết luận chương	70
KẾT LUẬN	72
DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ	76

DANH MỤC CÁC BẢNG

Bảng 3.3.1. Ký hiệu và tham số chính của mô hình hệ thống	47
Bảng 3.3.2. Phân loại VNF theo khả năng chia sẻ trong mô hình	47
Bảng 3.4.1. Trọng số sử dụng trong hàm điểm của DWSA-VNF	51
Bảng 3.4.2. Các thành phần của cơ chế chấm điểm trong DWSA-VNF	51
Bảng 4.2.1. Cấu hình chính của môi trường mô phỏng	59
Bảng 4.2.2. Các phương pháp được sử dụng trong so sánh thực nghiệm . . .	60
Bảng 4.4.1. Tóm tắt so sánh theo khả năng phục vụ và chất lượng dịch vụ . .	66
Bảng 4.5.1. Tóm tắt so sánh theo hiệu quả sử dụng tài nguyên và chi phí vận hành	69

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 3.4.1. Pipeline tổng thể của phương pháp DWSA-VNF	49
Hình 4.4.1. So sánh <i>tỷ lệ chấp nhận yêu cầu</i> (acceptance ratio) giữa các phương pháp placement	63
Hình 4.4.2. So sánh <i>thông lượng mạng</i> (network throughput) giữa các phương pháp placement	64
Hình 4.4.3. So sánh <i>độ trễ đầu cuối trung bình</i> (average end-to-end delay) giữa các phương pháp placement	65
Hình 4.5.1. So sánh <i>mức sử dụng bộ xử lý trung tâm</i> (CPU utilization) giữa các phương pháp placement	67
Hình 4.5.2. So sánh <i>mức sử dụng bộ nhớ</i> (memory utilization) giữa các phương pháp placement	67
Hình 4.5.3. So sánh <i>mức sử dụng băng thông</i> (bandwidth utilization) giữa các phương pháp placement	68
Hình 4.5.4. So sánh <i>thời gian thực thi</i> (execution time) giữa các phương pháp placement	69

MỞ ĐẦU

Lý do chọn đề tài

Trong những năm gần đây, cùng với sự phát triển nhanh chóng của điện toán đám mây, mạng 5G, điện toán biên và các dịch vụ số quy mô lớn, hạ tầng mạng truyền thống dựa trên các thiết bị phần cứng chuyên dụng ngày càng bộc lộ nhiều hạn chế. Các chức năng mạng như *tường lửa* (Firewall), *cân bằng tải* (Load Balancer), *Dịch địa chỉ mạng* (Network Address Translation - NAT) hay *Hệ thống phát hiện xâm nhập* (Intrusion Detection System - IDS) nếu tiếp tục được triển khai dưới dạng middlebox độc lập sẽ làm tăng chi phí đầu tư, giảm tính linh hoạt khi mở rộng dịch vụ và gây khó khăn cho quá trình vận hành, bảo trì và nâng cấp hệ thống. Trong bối cảnh đó, *ảo hóa chức năng mạng* (Network Function Virtualization) được xem là một hướng phát triển quan trọng, cho phép chuyển các chức năng mạng từ thiết bị phần cứng chuyên dụng sang các thực thể phần mềm chạy trên hạ tầng tính toán phổ dụng, từ đó làm tăng tính linh hoạt, giảm chi phí và rút ngắn thời gian triển khai dịch vụ mới.

Tuy nhiên, NFV không chỉ đơn thuần là quá trình ảo hóa các chức năng mạng. Khi các *chức năng mạng ảo* (Virtual Network Function) được triển khai trên hạ tầng dùng chung và được liên kết thành các *chuỗi chức năng dịch vụ* (Service Function Chain), hiệu năng của toàn hệ thống phụ thuộc rất mạnh vào cách các VNF được đặt lên hạ tầng, được xâu chuỗi và được điều phối. Trong mạng NFV, *Placement* là việc quyết định đặt các VNF (Virtual Network Functions) ở đâu trong hạ tầng mạng/tài nguyên tính toán. Ví dụ: firewall ảo, load balancer, NAT, IDS sẽ chạy trên server nào, ở data center nào, hay ở edge/cloud nào. *Chaining* là việc nối các VNF theo một thứ tự xử lý dịch vụ, còn gọi là Service Function Chaining (SFC). Tức là lưu lượng đi qua chuỗi các chức năng mạng theo thứ tự mong muốn. Một phương án placement không hợp lý có thể làm tăng *độ trễ đầu cuối* (end-to-end delay), giảm *thông lượng mạng* (network throughput), làm suy giảm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), đồng thời gây lãng phí tài nguyên tính toán và tài nguyên mạng. Do đó, placement/chaining trở thành một trong những bài toán trung tâm của môi trường NFV hiện đại.

Về tình hình nghiên cứu, nhiều công trình trong và ngoài nước đã tập trung vào các hướng như *phân bổ tài nguyên* (resource allocation), *đặt VNF* (VNF placement), *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), *điều hướng lưu lượng*

(traffic steering), tối ưu hiệu năng nền tảng và tăng độ tin cậy của hệ thống. Các hướng tiếp cận này bao gồm mô hình tối ưu hóa chính xác, heuristic, metaheuristic, cũng như các mô hình học máy và học tăng cường. Bên cạnh đó, các nghiên cứu gần đây cũng cho thấy vai trò ngày càng quan trọng của các yếu tố như *tải biến thiên theo thời gian* (time-varying workload), *đa thuê* (multi-tenancy), *dùng chung VNF* và *phụ tải tài nguyên nền* (Fundamental Resource Overhead) trong việc cải thiện hiệu năng hệ thống NFV.

Mặc dù vậy, vẫn còn một số vấn đề đặt ra. Thứ nhất, nhiều mô hình placement hiện có khá nặng về tối ưu hóa hình thức, dẫn đến chi phí tính toán cao và khó hiện thực trong môi trường nghiên cứu ứng dụng. Thứ hai, không phải công trình nào cũng xét đồng thời các yếu tố quan trọng như delay, workload động và cơ chế dùng chung VNF một cách hợp lý. Thứ ba, một số hướng chia sẻ VNF còn mang tính toàn phần, chưa phản ánh đầy đủ sự khác biệt giữa các loại chức năng mạng trong thực tế. Từ những vấn đề đó, việc nghiên cứu công nghệ NFV và đề xuất một phương pháp cải tiến hiệu năng theo hướng heuristic, đủ đơn giản để hiện thực nhưng vẫn có cơ sở kỹ thuật rõ ràng, là một hướng nghiên cứu có ý nghĩa cả về học thuật lẫn thực tiễn.

Xuất phát từ các lý do trên, học viên lựa chọn đề tài: “Nghiên cứu công nghệ ảo hóa chức năng mạng (NFV) và đề xuất phương pháp cải tiến hiệu năng”.

Mục đích nghiên cứu

Mục đích nghiên cứu của luận văn là nghiên cứu nền tảng công nghệ *ảo hóa chức năng mạng* (Network Function Virtualization), phân tích các hướng cải tiến hiệu năng đã được công bố, từ đó đề xuất một phương pháp placement/chaining phù hợp với đặc điểm của môi trường NFV nhằm nâng cao hiệu quả khai thác hạ tầng. Trên cơ sở đó, luận văn xây dựng một mô hình mô phỏng thống nhất để kiểm chứng và đánh giá định lượng hiệu quả của phương pháp đề xuất.

Cụ thể hơn, luận văn hướng tới việc xây dựng một giải pháp đặt chỗ (placement) có xét đồng thời *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing), qua đó cải thiện khả năng phục vụ dịch vụ, nâng cao hiệu quả sử dụng tài nguyên và duy trì chất lượng dịch vụ ở mức phù hợp.

Nội dung nghiên cứu

Để đạt được mục đích trên, luận văn tập trung thực hiện các nội dung nghiên cứu chủ yếu sau đây.

Thứ nhất, nghiên cứu tổng quan về công nghệ NFV, SDN và SFC, làm rõ kiến trúc và các thành phần chính của hệ thống NFV, đồng thời phân tích các yêu cầu kỹ thuật và các chỉ tiêu hiệu năng trong môi trường NFV/SFC.

Thứ hai, khảo sát và hệ thống hóa các nhóm phương pháp cải tiến hiệu năng NFV đã được nghiên cứu, đặc biệt là các hướng liên quan đến resource allocation, service orchestration, VNF placement, chaining, workload động, multi-tenancy và dùng chung VNF instance.

Thứ ba, phân tích các hướng nghiên cứu gần với đề tài để xác định khoảng trống nghiên cứu, từ đó làm cơ sở cho việc đề xuất phương pháp cải tiến hiệu năng phù hợp với bối cảnh của luận văn.

Thứ tư, xây dựng mô hình hệ thống NFV phục vụ bài toán placement/chaining, xác định các giả thiết mô phỏng, thiết kế heuristic DWSA-VNF theo hướng xét đồng thời delay, workload theo thời gian và selective sharing.

Thứ năm, hiện thực môi trường mô phỏng bằng lập trình Python, thực hiện so sánh phương pháp đề xuất với các baseline như Random Placement, Greedy First-Fit và Greedy Share-All.

Thứ sáu, đánh giá kết quả mô phỏng thông qua các chỉ số như *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối trung bình* (average end-to-end delay), *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization) và *thời gian thực thi* (execution time).

Cơ sở khoa học và tính thực tiễn của đề tài

Cơ sở khoa học của đề tài được xây dựng trên nền tảng các lý thuyết và kết quả nghiên cứu về NFV, SDN, SFC, resource allocation, service orchestration, VNF placement và các hướng tối ưu hiệu năng trong môi trường ảo hóa mạng. Các khảo sát và công trình nghiên cứu gần đây cho thấy placement/chaining là một trong những bài toán quan trọng nhất quyết định hiệu năng của hệ thống NFV. Đồng thời, các tài liệu cũng chỉ ra rằng những yếu tố như delay, workload biến thiên theo thời gian, dùng chung VNF và hiệu quả khai thác tài nguyên có ảnh hưởng trực tiếp đến chất lượng dịch vụ và hiệu quả vận hành của hệ thống. Trên cơ sở đó, luận văn lựa chọn hướng tiếp cận heuristic nhằm tạo ra một phương pháp vừa có cơ sở khoa học rõ ràng, vừa có khả năng hiện thực trong một môi trường nghiên cứu ứng dụng.

Tính thực tiễn của đề tài thể hiện ở chỗ NFV đang là xu hướng phát triển tất yếu của hạ tầng mạng hiện đại, đặc biệt trong các hệ thống đòi hỏi mức linh hoạt cao, khả năng mở rộng nhanh và chi phí vận hành hợp lý. Bài toán placement/chaining trong môi trường NFV có ý nghĩa thực tiễn rõ rệt vì nó ảnh hưởng trực tiếp tới hiệu quả khai thác hạ tầng, khả năng phục vụ dịch vụ và mức độ thỏa mãn các yêu cầu SLA. Phương pháp đề xuất trong luận văn được xây dựng theo hướng kỹ thuật ứng dụng, có thể hiện thực bằng môi trường lập trình Python, để kiểm chứng bằng mô phỏng và có thể làm nền cho các nghiên cứu mở rộng sau này.

Những đóng góp của luận văn

Những đóng góp chính của luận văn có thể được tóm tắt như sau.

Thứ nhất, luận văn đã cố gắng hệ thống hóa tương đối đầy đủ cơ sở lý thuyết liên quan đến NFV, SDN, SFC, kiến trúc hệ thống NFV và các nhóm phương pháp cải tiến hiệu năng trong môi trường ảo hóa mạng, qua đó làm rõ vị trí của bài toán placement/chaining trong tổng thể nghiên cứu về NFV.

Thứ hai, luận văn đã phân tích các hướng nghiên cứu gần với đề tài, đặc biệt là các hướng placement có xét delay, workload động, multi-tenancy và dùng chung VNF, từ đó xác định được khoảng trống nghiên cứu phù hợp với phạm vi của một luận văn thạc sĩ.

Thứ ba, luận văn đã đề xuất phương pháp DWSA-VNF như một kỹ thuật placement/chaining có xét đồng thời *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing).

Thứ tư, luận văn đã xây dựng một môi trường mô phỏng thống nhất sử dụng ngôn ngữ Python, cho phép hiện thực các baseline và phương pháp đề xuất trong cùng một pipeline xử lý.

Thứ năm, luận văn đã đánh giá định lượng hiệu quả của phương pháp đề xuất thông qua các chỉ số thực nghiệm quan trọng, từ đó cho thấy tính khả thi và ý nghĩa của phương pháp DWSA-VNF trong bối cảnh placement/chaining của môi trường NFV.

Đối tượng nghiên cứu

Đối tượng nghiên cứu của luận văn là công nghệ *ảo hóa chức năng mạng* (Network Function Virtualization) và bài toán cải tiến hiệu năng trong môi trường NFV, với trọng tâm là bài toán *đặt VNF* (VNF placement) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining). Trong phạm vi đó, luận văn tập trung nghiên

cứu cách các *chức năng mạng ảo* (Virtual Network Function) được triển khai trên hạ tầng dùng chung, được liên kết thành các chuỗi dịch vụ và được điều phối sao cho đạt hiệu quả phục vụ tốt hơn.

Đối tượng nghiên cứu trực tiếp hơn của luận văn là các yếu tố có ảnh hưởng mạnh đến hiệu năng placement/chaining, bao gồm *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload), cơ chế dùng chung VNF instance, hiệu quả sử dụng tài nguyên tính toán và tài nguyên mạng, cùng với các chỉ số đánh giá như acceptance ratio, network throughput, average end-to-end delay, CPU utilization, memory utilization, bandwidth utilization và execution time.

Phương pháp nghiên cứu

Luận văn sử dụng kết hợp nhiều phương pháp nghiên cứu nhằm bảo đảm tính chặt chẽ về mặt học thuật và tính khả thi về mặt triển khai.

Thứ nhất là phương pháp Khảo sát, phân tích và tổng hợp tài liệu. Phương pháp này được sử dụng để nghiên cứu các công trình liên quan đến NFV, SDN, SFC, phân bổ tài nguyên (resource allocation), điều phối dịch vụ (service orchestration), placement/chaining và các hướng tối ưu hiệu năng trong môi trường ảo hóa mạng, từ đó xây dựng cơ sở lý thuyết và xác định khoảng trống nghiên cứu.

Thứ hai là phương pháp Mô hình hóa hệ thống. Trên cơ sở tài liệu khảo sát, luận văn xây dựng mô hình hệ thống NFV dưới dạng hạ tầng hữu hạn tài nguyên, tập yêu cầu chuỗi dịch vụ, profile workload theo thời gian và cơ chế selective sharing, làm nền cho bài toán placement/chaining.

Thứ ba là phương pháp Thiết kế heuristic. Luận văn không lựa chọn tiếp cận theo mô hình tối ưu hóa chính xác có độ phức tạp cao, mà xây dựng heuristic DWSA-VNF theo hướng kỹ thuật ứng dụng, phù hợp với môi trường nghiên cứu thạc sĩ và có thể hiện thực hóa hiệu quả.

Thứ tư là phương pháp Mô phỏng và đánh giá thực nghiệm. Luận văn hiện thực toàn bộ môi trường mô phỏng bằng ngôn ngữ lập trình Python, chạy các baseline và phương pháp đề xuất trong cùng một pipeline xử lý, sau đó đánh giá kết quả theo các chỉ số định lượng đã xác định.

Sự kết hợp giữa các phương pháp trên giúp luận văn phát triển theo một mạch logic thống nhất, từ khảo sát tài liệu, đến mô hình hóa, thiết kế phương pháp và cuối cùng là kiểm chứng bằng thực nghiệm.

CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ ẢO HÓA CHỨC NĂNG MẠNG (NFV)

1.1. Đặt vấn đề

Trong những năm gần đây, sự phát triển nhanh chóng của các dịch vụ số, điện toán đám mây, mạng di động thế hệ mới và các hệ thống kết nối quy mô lớn đã làm thay đổi sâu sắc yêu cầu đối với hạ tầng mạng. Nếu trong các thế hệ trước, các chức năng mạng như *tường lửa* (firewall), *cân bằng tải* (load balancer), *hệ thống phát hiện xâm nhập* (intrusion detection system) hay *dịch địa chỉ mạng* (network address translation) thường được triển khai dưới dạng các thiết bị phần cứng chuyên dụng, thì trong bối cảnh hiện nay, mô hình đó dần bộc lộ nhiều hạn chế. Những hạn chế dễ nhận thấy gồm chi phí đầu tư lớn, mức độ linh hoạt thấp, khó mở rộng khi nhu cầu dịch vụ tăng nhanh, và khó thích ứng khi phải thường xuyên đưa ra các dịch vụ mới trong thời gian ngắn [1, 2].

Xuất phát từ nhu cầu mềm hóa hạ tầng mạng và tăng cường khả năng tự động hóa vận hành, *ảo hóa chức năng mạng* (Network Function Virtualization) và *mạng điều khiển bằng phần mềm* (Software-Defined Networking) đã trở thành hai công nghệ nền tảng của các hệ thống mạng hiện đại. NFV cho phép chuyển các chức năng mạng từ các thiết bị phần cứng riêng lẻ sang các thực thể phần mềm có thể chạy trên hạ tầng tính toán phổ dụng, trong khi SDN cung cấp khả năng tách biệt *mặt phẳng điều khiển* (control plane) khỏi *mặt phẳng dữ liệu* (data plane), qua đó hỗ trợ điều hướng lưu lượng và quản trị mạng theo hướng linh hoạt hơn [1, 3, 4]. Khi kết hợp lại, hai công nghệ này tạo ra nền tảng cho việc xây dựng, triển khai và điều phối các dịch vụ mạng theo hướng linh hoạt, động và có thể lập trình được.

Một bước phát triển quan trọng tiếp theo là sự xuất hiện của *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), trong đó các *chức năng mạng ảo* (Virtual Network Function) được liên kết thành một chuỗi xử lý dịch vụ phù hợp với từng loại lưu lượng hoặc từng yêu cầu người dùng. Theo cách tiếp cận này, một luồng dữ liệu không chỉ đơn thuần được truyền từ nguồn tới đích, mà còn phải đi qua một tập các chức năng mạng theo đúng thứ tự, chẳng hạn firewall, IDS, NAT hoặc load balancer, tùy theo chính sách dịch vụ cụ thể [3, 5, 6]. Điều này làm thay đổi căn bản bài toán vận hành mạng: thay vì chỉ quản lý từng thiết bị riêng lẻ, hệ thống phải quản lý đồng

thời placement của các VNF, logic chaining giữa chúng và cách lưu lượng được điều khiển qua chuỗi dịch vụ.

Tuy mang lại nhiều lợi ích rõ rệt về linh hoạt, tiết kiệm chi phí và khả năng mở rộng, môi trường NFV cũng đặt ra nhiều thách thức kỹ thuật mới. Khi nhiều VNF cùng chạy trên một hạ tầng chia sẻ, hệ thống phải quyết định mỗi chức năng được đặt ở đâu, sử dụng bao nhiêu tài nguyên, và làm thế nào để lưu lượng đi qua đúng chuỗi xử lý mà vẫn bảo đảm *chất lượng dịch vụ* (Quality of Service). Các quyết định này ảnh hưởng trực tiếp đến *độ trễ đầu cuối* (end-to-end delay), *thông lượng mạng* (network throughput), *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), cũng như mức sử dụng tài nguyên tính toán và tài nguyên mạng của toàn hệ thống [7–9]. Nói cách khác, hiệu năng của môi trường NFV không phải là hệ quả tự nhiên của việc ảo hóa, mà phụ thuộc rất mạnh vào chất lượng của các quyết định placement, chaining và orchestration.

Từ góc độ nghiên cứu, điều này khiến bài toán cải tiến hiệu năng trong NFV trở thành một hướng rất được quan tâm. Nhiều công trình đã đề xuất các phương pháp tối ưu hóa hoặc heuristic để giải quyết những vấn đề như *phân bổ tài nguyên* (resource allocation), *đặt VNF* (VNF placement), *điều hướng lưu lượng* (traffic steering), *đa thuê* (multi-tenancy), *tối ưu hiệu năng thực thi* (performance tuning) và *bảo đảm độ tin cậy dịch vụ* (service resilience) [1, 3, 7, 10]. Trong số đó, placement/chaining nổi lên như một trong những nội dung trung tâm nhất, vì đây là nơi các yếu tố về tài nguyên, độ trễ, cấu trúc dịch vụ và logic điều phối gặp nhau một cách trực tiếp.

Đối với luận văn này, bài toán được quan tâm không chỉ là việc đặt được các VNF lên hạ tầng, mà là đặt chúng như thế nào để cải thiện hiệu năng tổng thể của hệ thống. Cụ thể hơn, luận văn quan tâm đến một số yếu tố đang được xem là quan trọng trong các nghiên cứu gần đây, bao gồm *tải biến thiên theo thời gian* (time-varying workload), khả năng dùng chung instance giữa các dịch vụ, và yêu cầu duy trì độ trễ đầu cuối ở mức phù hợp. Những yếu tố này làm cho bài toán placement trở nên khó hơn, nhưng đồng thời cũng mở ra cơ hội để xây dựng các heuristic có chất lượng tốt hơn so với các phương pháp tham lam đơn giản [11–13].

Từ những phân tích trên có thể thấy rằng việc nghiên cứu công nghệ NFV và đề xuất một phương pháp cải tiến hiệu năng phù hợp không chỉ có ý nghĩa học thuật, mà còn mang giá trị thực tiễn rõ rệt đối với vận hành dịch vụ mạng hiện đại. Đây chính là cơ sở để luận văn lựa chọn đề tài “*Nghiên cứu công nghệ ảo hóa chức năng mạng*”

(NFV) và đề xuất phương pháp cải tiến hiệu năng”. Trên nền tảng vấn đề đã được đặt ra ở mục này, phần tiếp theo sẽ làm rõ hơn tính cấp thiết của đề tài, qua đó xác định cụ thể lý do vì sao bài toán placement/chaining trong môi trường NFV cần tiếp tục được nghiên cứu theo hướng kỹ thuật ứng dụng.

1.2. Tính cấp thiết của đề tài

Sự cấp thiết của đề tài trước hết xuất phát từ chính xu hướng phát triển của hạ tầng mạng và dịch vụ số hiện nay. Trong bối cảnh lưu lượng dữ liệu tăng nhanh, mô hình dịch vụ ngày càng đa dạng và yêu cầu thời gian triển khai ngày càng ngắn, các hệ thống mạng cần một kiến trúc đủ linh hoạt để thích ứng với thay đổi liên tục của nhu cầu vận hành. NFV được xem là một lời giải phù hợp vì cho phép triển khai các chức năng mạng dưới dạng phần mềm, từ đó hỗ trợ mở rộng hoặc tái cấu hình dịch vụ nhanh hơn rất nhiều so với mô hình thiết bị phần cứng truyền thống [1, 2]. Tuy nhiên, tính linh hoạt mà NFV mang lại cũng đồng thời làm tăng độ phức tạp của việc điều phối tài nguyên và tối ưu hiệu năng.

Trong một môi trường NFV thực tế, hạ tầng thường là hữu hạn, trong khi số lượng dịch vụ cần triển khai có thể lớn và thay đổi theo thời gian. Mỗi dịch vụ lại có thể bao gồm một *chuỗi chức năng dịch vụ* (Service Function Chain) riêng, yêu cầu đi qua các VNF theo đúng thứ tự và trong giới hạn *cam kết mức dịch vụ* (Service Level Agreement). Điều này có nghĩa là cùng một hạ tầng vật lý, các cách placement khác nhau có thể dẫn tới các kết quả rất khác nhau về *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối* (end-to-end delay) và mức sử dụng tài nguyên [7–9]. Nói cách khác, nếu placement không hợp lý, lợi ích của NFV sẽ bị suy giảm đáng kể, thậm chí hệ thống có thể hoạt động kém hiệu quả hơn mong đợi dù hạ tầng vật lý không hề nhỏ.

Một yếu tố làm tăng thêm tính cấp thiết của bài toán là đặc tính động của tải hệ thống. Trong nhiều trường hợp, các yêu cầu dịch vụ không duy trì ở một mức tải cố định, mà biến đổi theo thời gian tùy thuộc vào hành vi người dùng, thời điểm trong ngày hoặc đặc trưng của ứng dụng. Nếu các quyết định placement chỉ dựa trên một mức tải tĩnh, hệ thống rất dễ gặp tình trạng phân bổ tài nguyên chưa tối ưu, dẫn đến một số node quá tải trong khi các node khác vẫn còn dư tài nguyên. Các nghiên cứu gần đây cho thấy việc đưa *tải biến thiên theo thời gian* (time-varying workload) vào quá trình ra quyết định có thể giúp cải thiện đáng kể hiệu quả khai thác hạ tầng [11, 14]. Điều này cho thấy cải tiến hiệu năng trong NFV không thể chỉ dừng lại ở các mô hình

placement tĩnh đơn giản.

Một khía cạnh quan trọng khác là bài toán dùng chung VNF instance giữa nhiều dịch vụ. Về nguyên tắc, nếu mọi yêu cầu đều khởi tạo instance riêng cho từng VNF, hệ thống có thể tiêu tốn quá nhiều tài nguyên cho các *phụ tải tài nguyên nền* (Fundamental Resource Overhead) thay vì cho phần xử lý dịch vụ thực sự. Tuy nhiên, cơ chế dùng chung cũng không thể áp dụng đồng loạt cho mọi loại VNF, bởi một số chức năng gắn với trạng thái và chính sách riêng. Điều này cho thấy môi trường NFV hiện đại đòi hỏi những phương pháp placement/chaining tinh tế hơn, không chỉ biết tận dụng sharing khi có lợi, mà còn biết giới hạn sharing trong các trường hợp cần thiết [1, 11, 12].

Từ góc độ nghiên cứu học thuật, sự cấp thiết của đề tài còn thể hiện ở chỗ placement/chaining vẫn là một bài toán mở với nhiều hướng tiếp cận khác nhau. Một số công trình lựa chọn mô hình tối ưu hóa chính xác, trong khi một số khác chuyển sang heuristic, approximation hoặc học tăng cường để giảm chi phí tính toán. Tuy nhiên, nhiều hướng tiếp cận hoặc còn quá nặng về mô hình toán học, hoặc khó hiện thực trong một môi trường nghiên cứu ứng dụng gọn nhẹ, hoặc chưa xét đồng thời các yếu tố như delay, workload động và selective sharing trong cùng một khung giải quyết [8, 9, 13, 15]. Điều này mở ra nhu cầu nghiên cứu một phương pháp vừa đủ đơn giản để triển khai, vừa đủ chặt chẽ để có ý nghĩa học thuật và có thể được đánh giá định lượng trong môi trường mô phỏng.

Đối với nội dung của một luận văn thạc sĩ, đây là một yêu cầu rất phù hợp. Đề tài không đòi hỏi phải giải quyết toàn bộ mọi vấn đề của NFV, nhưng cần thể hiện được khả năng lựa chọn đúng một hướng nghiên cứu có ý nghĩa, khảo sát được nền tảng học thuật liên quan, và đề xuất được một phương pháp cải tiến hiệu năng có cơ sở kỹ thuật rõ ràng. Trong trường hợp này, việc tập trung vào bài toán placement/chaining có xét delay, workload theo thời gian và selective sharing là một lựa chọn phù hợp, vừa bám sát xu hướng nghiên cứu hiện nay, vừa có khả năng hiện thực trong môi trường lập trình Python để kiểm chứng bằng mô phỏng. Ngoài ra, tính cấp thiết của đề tài còn thể hiện ở khoảng cách giữa nghiên cứu lý thuyết và khả năng hiện thực trong thực hành. Nhiều mô hình placement trong tài liệu được xây dựng rất chặt chẽ về mặt tối ưu hóa, nhưng khi chuyển sang môi trường mô phỏng hoặc môi trường triển khai gần thực tế, chi phí tính toán, số lượng biến quyết định và số lượng ràng buộc thường tăng rất nhanh theo quy mô topology và số lượng yêu cầu dịch vụ [8, 9]. Điều này tạo ra nhu cầu đối với các phương pháp heuristic hoặc gần tối ưu, tức các phương pháp chấp

nhận đánh đổi một phần tính tối ưu hình thức để đổi lấy khả năng triển khai, khả năng lặp lại thí nghiệm và khả năng mở rộng trên các kịch bản thực nghiệm đa dạng hơn. Trong bối cảnh của một luận văn thạc sĩ theo hướng ứng dụng, đây là một định hướng phù hợp, vì nó cho phép tập trung vào việc xây dựng một phương pháp có logic kỹ thuật rõ ràng, có thể hiện thực bằng công cụ quen thuộc và có thể đánh giá định lượng trong môi trường mô phỏng thống nhất.

Như vậy, tính cấp thiết của đề tài có thể được nhìn nhận ở cả ba phương diện. Thứ nhất, về phương diện công nghệ, NFV là xu hướng tất yếu của hạ tầng mạng hiện đại và đòi hỏi các giải pháp điều phối hiệu quả hơn. Thứ hai, về phương diện kỹ thuật, placement/chaining có ảnh hưởng trực tiếp đến các chỉ số hiệu năng quan trọng của hệ thống. Thứ ba, về phương diện nghiên cứu, vẫn còn khoảng trống giữa các hướng tối ưu hóa phức tạp và nhu cầu xây dựng một heuristic thực dụng, phù hợp với điều kiện triển khai và đánh giá của một luận văn ứng dụng. Chính từ sự hội tụ của ba phương diện này mà đề tài nghiên cứu được lựa chọn và phát triển trong luận văn.

Trên cơ sở tính cấp thiết đã được làm rõ, các mục tiếp theo của Chương 1 sẽ trình bày cụ thể mục tiêu, đối tượng, phạm vi và phương pháp nghiên cứu của luận văn. Đồng thời, những nội dung đó cũng sẽ tạo ra cầu nối trực tiếp sang Chương 2, nơi các công nghệ NFV, SDN, SFC và các nhóm phương pháp cải tiến hiệu năng sẽ được khảo sát một cách hệ thống hơn.

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

Trên cơ sở bối cảnh và tính cấp thiết của đề tài đã trình bày, luận văn xác định mục tiêu nghiên cứu tổng quát là khảo sát công nghệ *ảo hóa chức năng mạng* (Network Function Virtualization), phân tích các hướng cải tiến hiệu năng đã được nghiên cứu, từ đó đề xuất một phương pháp phù hợp để nâng cao hiệu quả khai thác hạ tầng trong môi trường NFV. Mục tiêu này không dừng lại ở việc tổng hợp lý thuyết, mà hướng tới việc chuyển hóa phần khảo sát thành một phương pháp đề xuất có thể hiện thực và đánh giá được trong môi trường mô phỏng thống nhất [3, 7, 11].

Để cụ thể hóa mục tiêu tổng quát nói trên, luận văn đặt ra một số mục tiêu thành phần. Thứ nhất, luận văn nghiên cứu nền tảng công nghệ của NFV, SDN và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), đồng thời làm rõ mối quan hệ giữa các công nghệ này trong việc triển khai dịch vụ mạng hiện đại. Thứ hai, luận văn khảo sát các nhóm phương pháp cải tiến hiệu năng trong môi trường NFV, đặc biệt là các hướng liên quan trực tiếp đến *đặt VNF* (VNF placement), phân bổ tài nguyên,

điều phối dịch vụ và tối ưu hiệu năng hệ thống [1, 8, 9]. Thứ ba, luận văn phân tích các hướng tiếp cận gần với đề tài, chẳng hạn các phương pháp có xét *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload), *đa thuê* (multi-tenancy) và cơ chế dùng chung VNF instance. Trên cơ sở đó, luận văn đề xuất một heuristic placement mang tên DWSA-VNF, và cuối cùng đánh giá phương pháp này bằng mô phỏng trong môi trường lập trình Python.

Về *đối tượng nghiên cứu*, luận văn tập trung vào môi trường NFV trong đó các *chức năng mạng ảo* (Virtual Network Function) được triển khai trên hạ tầng dùng chung và được liên kết thành các chuỗi chức năng dịch vụ. Trong không gian đó, đối tượng nghiên cứu trực tiếp của luận văn là bài toán placement/chaining, tức bài toán xác định mỗi VNF nên được đặt ở đâu, liên kết ra sao với các VNF khác trong chuỗi dịch vụ, và cách nào để việc ánh xạ đó giúp hệ thống hoạt động hiệu quả hơn. Bên cạnh đó, luận văn cũng quan tâm tới một số chỉ tiêu hiệu năng gắn trực tiếp với bài toán này, bao gồm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối trung bình* (average end-to-end delay), *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization) và *thời gian thực thi* (execution time) [7, 11, 16].

Về *phạm vi nghiên cứu*, luận văn giới hạn trọng tâm vào bài toán placement/chaining trong môi trường NFV có xét một số yếu tố được xem là quan trọng đối với hiệu năng hệ thống. Cụ thể, phương pháp đề xuất được xây dựng với mục tiêu xử lý đồng thời ba yếu tố chính: ràng buộc delay, workload theo thời gian và cơ chế *chia sẻ có chọn lọc* (selective sharing). Việc chọn ba yếu tố này không phải ngẫu nhiên, mà dựa trên nhận định từ tài liệu rằng đây là những thành phần có ảnh hưởng rõ rệt đến hiệu quả placement trong NFV hiện đại [8, 11, 12]. Đồng thời, phạm vi nghiên cứu của luận văn chưa mở rộng sâu sang các chủ đề như backup, failover, migration, bảo mật nền tảng hoặc triển khai thực tế trên testbed vật lý, vì việc bao phủ tất cả các hướng đó sẽ vượt quá giới hạn của một luận văn thạc sĩ theo định hướng ứng dụng.

Một giới hạn khác của luận văn là môi trường kiểm chứng được xây dựng dưới dạng mô phỏng, thay vì triển khai trên hệ thống mạng thực tế quy mô lớn. Tuy nhiên, điều này không làm giảm ý nghĩa của nghiên cứu. Trên thực tế, mô phỏng là một cách tiếp cận phổ biến trong các nghiên cứu NFV placement vì cho phép kiểm soát topology, tập yêu cầu, profile workload và các chỉ số đánh giá trong một môi trường thống nhất [3, 8]. Quan trọng hơn, trong luận văn này, môi trường mô phỏng được

thực hiện bằng ngôn ngữ phổ biến Python, giúp bảo đảm tính nhất quán giữa mô hình hệ thống, phương pháp đề xuất và kết quả đánh giá ở Chương 4.

Từ những xác định trên có thể thấy rằng luận văn lựa chọn một phạm vi nghiên cứu vừa đủ hẹp để bảo đảm tính khả thi, nhưng vẫn đủ rộng để tạo ra đóng góp có ý nghĩa. Luận văn không tìm cách giải quyết toàn bộ bài toán tối ưu hiệu năng của NFV, mà tập trung vào một lát cắt có tính đại diện cao, đó là bài toán placement/chaining có xét delay, workload động và selective sharing. Chính nhờ sự lựa chọn phạm vi như vậy mà phần phương pháp ở Chương 3 có thể được xây dựng rõ ràng, còn phần đánh giá ở Chương 4 có thể bám sát đúng mục tiêu ban đầu của đề tài.

1.4. Phương pháp nghiên cứu

Để thực hiện các mục tiêu đã đặt ra, luận văn sử dụng một cách tiếp cận nghiên cứu kết hợp giữa khảo sát lý thuyết, phân tích hệ thống, thiết kế thuật toán và mô phỏng thực nghiệm. Cách tổ chức này phù hợp với bản chất của bài toán cải tiến hiệu năng trong môi trường NFV, bởi đây là một bài toán vừa có nền tảng học thuật rõ ràng, vừa đòi hỏi một phương pháp đủ cụ thể để có thể triển khai và đánh giá bằng thực nghiệm [1, 3, 7].

Phương pháp đầu tiên được sử dụng là *khảo sát, phân tích và tổng hợp tài liệu*. Đây là phương pháp nền tảng, được dùng để xây dựng cơ sở lý thuyết cho toàn bộ luận văn. Thông qua việc khảo sát các survey, paper và luận án liên quan, luận văn hệ thống hóa các khái niệm về NFV, SDN, SFC, kiến trúc NFV platform, các nhóm phương pháp cải tiến hiệu năng và các hướng nghiên cứu placement/chaining hiện nay. Việc phân tích tài liệu không chỉ nhằm tổng hợp kiến thức nền, mà còn nhằm xác định xu hướng nghiên cứu, các chỉ tiêu đánh giá thường dùng, những hạn chế còn tồn tại và khoảng trống mà luận văn có thể khai thác [1, 3, 8, 11]. Phương pháp này được thể hiện rõ nhất trong Chương 2, nơi luận văn tiến hành khảo sát có chọn lọc các hướng nghiên cứu liên quan trực tiếp đến đề tài.

Phương pháp thứ hai là *mô hình hóa hệ thống*. Trên cơ sở tài liệu khảo sát, luận văn xây dựng một mô hình hệ thống NFV đủ chặt chẽ để mô tả bài toán placement/chaining, nhưng không quá nặng về hình thức toán học. Cụ thể, hạ tầng được biểu diễn như một đồ thị gồm các node và liên kết hữu hạn tài nguyên; mỗi yêu cầu dịch vụ được biểu diễn dưới dạng một *chuỗi chức năng dịch vụ* (Service Function Chain) có ràng buộc về tài nguyên và độ trễ; workload được mô tả theo các *khoảng thời gian rời rạc* (time-slots); còn khả năng dùng chung VNF được mô hình hóa theo cơ chế

selective sharing. Mô hình này đóng vai trò trung gian giữa phần tổng quan lý thuyết và phần thiết kế thuật toán, bảo đảm rằng phương pháp đề xuất ở Chương 3 được xây dựng trên một nền tảng mô tả nhất quán.

Phương pháp thứ ba là *thiết kế heuristic*. Sau khi xác định khoảng trống nghiên cứu, luận văn không lựa chọn xây dựng một mô hình tối ưu hóa chính xác quá phức tạp, mà hướng tới việc thiết kế một heuristic placement có thể hiện thực hóa hiệu quả trong môi trường nghiên cứu ứng dụng. Cách tiếp cận này phù hợp với thực tế rằng nhiều bài toán placement trong NFV có độ phức tạp cao, khó giải tối ưu khi kích thước topology và số lượng yêu cầu tăng lên [8, 9]. Do đó, luận văn lựa chọn phát triển phương pháp DWSA-VNF theo hướng kết hợp các nguyên tắc kỹ thuật như sắp xếp yêu cầu, sàng lọc node ứng viên, kiểm tra độ trễ, xét workload theo thời gian và selective sharing. Việc thiết kế heuristic cho phép luận văn giữ được sự cân bằng giữa tính học thuật và khả năng triển khai.

Phương pháp thứ tư là *mô phỏng và đánh giá thực nghiệm*. Sau khi xây dựng phương pháp đề xuất, luận văn hiện thực toàn bộ môi trường mô phỏng bằng ngôn ngữ Python theo mô hình script *tất cả trong một* chứa tất cả các thông tin môi trường, kịch bản thực nghiệm mô phỏng, nhằm đảm bảo tính thống nhất. Trong môi trường này, topology vật lý, tập yêu cầu dịch vụ, workload theo thời gian, các baseline và phương pháp DWSA-VNF đều được xử lý trong cùng một pipeline. Cách tổ chức như vậy giúp các kết quả thu được có tính nhất quán cao và cho phép so sánh trực tiếp giữa các thuật toán theo cùng một tập chỉ số [11, 16]. Đây là phương pháp chủ đạo được sử dụng ở Chương 4, nơi luận văn đánh giá định lượng phương pháp đề xuất thông qua các chỉ số như acceptance ratio, network throughput, average end-to-end delay, CPU utilization, memory utilization, bandwidth utilization và execution time.

Một điểm cần nhấn mạnh là các phương pháp trên không tồn tại độc lập, mà được tổ chức thành một chuỗi logic xuyên suốt toàn bộ luận văn. khảo sát tài liệu tạo nền cho mô hình hóa hệ thống; mô hình hóa hệ thống làm cơ sở cho thiết kế heuristic; heuristic sau đó được hiện thực hóa trong môi trường mô phỏng; và kết quả mô phỏng được dùng để kiểm chứng mức độ phù hợp của phương pháp đã đề xuất. Cách tổ chức như vậy giúp luận văn phát triển theo một mạch nghiên cứu tương đối chặt chẽ, trong đó mỗi chương đều có vai trò rõ ràng và bổ trợ trực tiếp cho chương tiếp theo.

Từ cách tiếp cận này, có thể thấy luận văn không chỉ dừng lại ở việc mô tả công nghệ hay tổng hợp các công trình đã có, mà thực sự kết hợp giữa nghiên cứu lý thuyết

với kiểm chứng bằng thực nghiệm. Đây cũng chính là điểm phù hợp với định hướng của một luận văn thạc sĩ theo hướng ứng dụng: xây dựng một nền tảng học thuật đủ vững, nhưng đồng thời phải đưa ra được một phương pháp cụ thể và chứng minh được hiệu quả của phương pháp đó trong một môi trường đánh giá nhất quán. Trên cơ sở đó, các mục tiếp theo của Chương 1 sẽ làm rõ hơn ý nghĩa của đề tài và bố cục của toàn bộ luận văn. Cũng cần nhấn mạnh rằng việc kết hợp giữa khảo sát tài liệu, mô hình hóa hệ thống, thiết kế heuristic và mô phỏng thực nghiệm không chỉ là một lựa chọn thuận tiện, mà là một lựa chọn có tính phương pháp luận. Nếu chỉ dừng lại ở khảo sát tài liệu, luận văn sẽ khó tạo ra được đóng góp riêng. Nếu chỉ tập trung vào mô phỏng mà không có nền phân tích lý thuyết, kết quả thực nghiệm sẽ thiếu cơ sở học thuật để diễn giải. Tương tự, nếu xây dựng thuật toán mà không có một mô hình hệ thống đủ rõ, phương pháp đề xuất sẽ thiếu tính nhất quán với phần đánh giá. Chính vì vậy, luận văn lựa chọn một chuỗi phương pháp có tính nối tiếp: tổng quan để xác định khoảng trống, mô hình hóa để xác định đối tượng giải quyết, thiết kế heuristic để hiện thực hóa ý tưởng nghiên cứu, và mô phỏng để kiểm chứng hiệu quả của phương pháp trong các kịch bản định lượng cụ thể.

1.5. Ý nghĩa khoa học và thực tiễn của đề tài

Bên cạnh tính cấp thiết về mặt công nghệ và kỹ thuật, đề tài còn có ý nghĩa rõ rệt ở cả hai phương diện Khoa học và Thực tiễn. Việc làm rõ hai phương diện này là cần thiết, vì một luận văn nghiên cứu ứng dụng không chỉ cần chỉ ra bài toán có đáng quan tâm hay không, mà còn phải chứng minh được rằng kết quả nghiên cứu có đóng góp nhất định đối với tri thức chuyên môn và có khả năng hỗ trợ cho hoạt động triển khai hoặc đánh giá trong thực tế.

Về *ý nghĩa khoa học*, trước hết luận văn góp phần hệ thống hóa cơ sở lý thuyết liên quan đến *ảo hóa chức năng mạng* (Network Function Virtualization), *mạng điều khiển bằng phần mềm* (Software-Defined Networking) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining). Trong bối cảnh lĩnh vực NFV phát triển nhanh và có phạm vi rất rộng, việc tổng hợp một cách có chọn lọc các khái niệm nền, kiến trúc hệ thống, các nhóm phương pháp cải tiến hiệu năng và các hướng nghiên cứu placement/chaining có ý nghĩa quan trọng đối với việc hình thành một bức tranh nghiên cứu rõ ràng và có định hướng [1,3,8]. Phần tổng quan này không chỉ giúp định vị bài toán của luận văn trong bối cảnh chung, mà còn tạo cơ sở để xác định các yếu tố kỹ thuật thực sự quan trọng đối với môi trường NFV hiện đại.

Ý nghĩa khoa học thứ hai của đề tài nằm ở việc phân tích và chỉ ra một khoảng trống nghiên cứu phù hợp với phạm vi của luận văn. Qua khảo sát tài liệu, có thể thấy rằng nhiều công trình hiện nay hoặc thiên về mô hình tối ưu hóa chính xác có độ phức tạp cao, hoặc tập trung vào từng khía cạnh riêng lẻ như delay, workload động, multi-tenancy hay resource efficiency [8, 9, 11]. Trong khi đó, nhu cầu thực tế đặt ra là cần một hướng tiếp cận cân bằng hơn, vừa đủ đơn giản để hiện thực và đánh giá trong môi trường nghiên cứu ứng dụng, nhưng vẫn đủ chặt chẽ để phản ánh các đặc trưng cốt lõi của bài toán placement trong NFV. Luận văn góp phần vào hướng này bằng cách lựa chọn và kết hợp ba yếu tố có ý nghĩa lớn đối với hiệu năng hệ thống, gồm *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing).

Ý nghĩa khoa học tiếp theo của đề tài là việc xây dựng một heuristic placement có logic kỹ thuật rõ ràng thay vì chỉ dừng lại ở mức khảo sát tài liệu. Phương pháp DWSA-VNF được đề xuất trong luận văn không nhằm thay thế toàn bộ các hướng tối ưu hóa đã có, mà đóng vai trò như một hướng tiếp cận phù hợp với bối cảnh nghiên cứu thực tế ứng dụng. Việc xây dựng một heuristic như vậy có ý nghĩa ở chỗ nó chuyển phần tổng quan lý thuyết thành một phương pháp cụ thể, có thể mô tả, có thể hiện thực và có thể kiểm chứng bằng mô phỏng. Điều này giúp luận văn tránh được tình trạng chỉ dừng ở mức tổng hợp tài liệu, đồng thời tạo ra một cầu nối hợp lý giữa nghiên cứu lý thuyết và đánh giá thực nghiệm [7, 11].

Về ý nghĩa thực tiễn, đề tài trước hết cung cấp một cách nhìn có hệ thống về bài toán placement/chaining trong môi trường NFV, từ đó giúp người nghiên cứu hoặc người học có thể tiếp cận bài toán này theo một mạch logic rõ ràng hơn. Trong thực tế, placement là một nội dung rất dễ bị nhìn nhận một cách rời rạc nếu chỉ xét riêng tài nguyên, riêng delay hoặc riêng throughput. Luận văn cho thấy rằng hiệu năng trong NFV cần được nhìn như kết quả của nhiều yếu tố phối hợp, trong đó placement là một điểm hội tụ của tài nguyên hạ tầng, ràng buộc dịch vụ và logic điều phối. Cách nhìn này có ích không chỉ cho bản thân đề tài, mà còn cho các nghiên cứu mở rộng về orchestration, scaling hoặc edge deployment sau này.

Ý nghĩa thực tiễn thứ hai là đề tài xây dựng được một môi trường mô phỏng thống nhất bằng lập trình Python, cho phép hiện thực các baseline và phương pháp đề xuất trong cùng một pipeline xử lý. Điều này có giá trị thực tế ở chỗ toàn bộ quá trình từ khởi tạo topology, sinh yêu cầu dịch vụ, mô phỏng workload theo thời gian, chạy thuật toán cho đến xuất kết quả đều được thực hiện trong một chương trình viết bằng

ngôn ngữ Python. Nhờ đó, kết quả mô phỏng có tính nhất quán cao hơn, dễ tái lập hơn và thuận lợi hơn cho việc trích xuất bảng số liệu, hình vẽ phục vụ phân tích [11]. Trong bối cảnh của một luận văn ứng dụng, đây là một đóng góp có tính thực hành rõ rệt.

Một ý nghĩa thực tiễn khác của đề tài là tạo ra một nền khởi đầu thuận lợi cho các nghiên cứu mở rộng tiếp theo trong lĩnh vực placement/chaining của môi trường NFV. Mặc dù luận văn chưa mở rộng sang các hướng như backup, failover, migration, security hardening hay triển khai trên testbed thực tế, heuristic DWSA-VNF và môi trường mô phỏng đi kèm vẫn có thể được dùng làm nền để bổ sung các yếu tố đó về sau. Nói cách khác, đóng góp thực tiễn của luận văn không chỉ nằm ở kết quả hiện tại, mà còn nằm ở việc tạo ra một khung nghiên cứu gọn, rõ và có khả năng mở rộng.

Ở một góc độ khác, ý nghĩa thực tiễn của đề tài còn nằm ở việc tạo ra một cách tiếp cận có tính sư phạm và có thể tái sử dụng. Bằng cách tổ chức bài toán placement trong một khung mô phỏng rõ ràng, luận văn không chỉ phục vụ cho mục tiêu bảo vệ một kết quả nghiên cứu cụ thể, mà còn tạo ra một nền tham chiếu thuận tiện cho các nghiên cứu kế tiếp. Những người nghiên cứu sau có thể thay đổi topology, bộ tham số, tập workload, nhóm VNF chia sẻ được hoặc các baseline so sánh mà không cần xây dựng lại toàn bộ môi trường từ đầu. Điều này làm tăng giá trị sử dụng của kết quả luận văn vượt ra ngoài phạm vi của một bài toán đơn lẻ, và góp phần nâng cao tính thực hành của nghiên cứu trong lĩnh vực NFV.

Từ các phân tích trên có thể thấy rằng đề tài vừa có ý nghĩa khoa học trong việc hệ thống hóa và phát triển một hướng tiếp cận placement phù hợp với đặc điểm NFV, vừa có ý nghĩa thực tiễn trong việc xây dựng một môi trường đánh giá thống nhất và một heuristic có thể kiểm chứng được. Chính sự kết hợp giữa hai phương diện này làm cho đề tài trở nên phù hợp với tính chất của một luận văn thạc sĩ theo hướng ứng dụng, nơi yêu cầu đặt ra không chỉ là nắm vững lý thuyết, mà còn phải thể hiện được khả năng đề xuất và kiểm chứng một giải pháp cụ thể.

1.6. Bố cục của luận văn

Để thực hiện các mục tiêu nghiên cứu đã nêu, luận văn được tổ chức thành bốn chương với mối liên hệ chặt chẽ và phát triển theo một mạch logic xuyên suốt từ bối cảnh, cơ sở lý thuyết đến phương pháp đề xuất và đánh giá thực nghiệm.

Chương 1 trình bày bối cảnh nghiên cứu, tính cấp thiết của đề tài, mục tiêu, đối tượng, phạm vi và phương pháp nghiên cứu, đồng thời làm rõ ý nghĩa khoa học và

thực tiễn của luận văn. Đây là chương đặt nền cho toàn bộ công trình, giúp xác định rõ bài toán cần giải quyết và hướng tiếp cận mà luận văn lựa chọn.

Chương 2 tập trung khảo sát cơ sở lý thuyết và các hướng nghiên cứu liên quan đến cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization). Nội dung chương bao gồm tổng quan về NFV, SDN và SFC; kiến trúc và các thành phần chính của hệ thống NFV; các yêu cầu kỹ thuật và chỉ tiêu hiệu năng trong môi trường NFV/SFC; các nhóm phương pháp cải tiến hiệu năng; cũng như phân tích các hướng nghiên cứu gần với đề tài để chỉ ra khoảng trống nghiên cứu. Chương này đóng vai trò làm nền học thuật trực tiếp cho phần phương pháp ở chương tiếp theo.

Chương 3 là chương trung tâm của luận văn, trong đó phương pháp DWSA-VNF được đề xuất và mô tả chi tiết. Trên cơ sở khoảng trống nghiên cứu đã xác định ở Chương 2, luận văn xây dựng một heuristic placement/chaining có xét đồng thời *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing). Bên cạnh phần mô hình hệ thống và thiết kế thuật toán, chương này còn trình bày môi trường cài đặt và tổ chức mô phỏng, qua đó tạo nền trực tiếp cho phần đánh giá thực nghiệm.

Chương 4 trình bày quá trình mô phỏng và đánh giá kết quả của phương pháp đề xuất. Trong chương này, luận văn thiết lập môi trường thí nghiệm, xác định các chỉ số đánh giá, so sánh DWSA-VNF với các baseline như Random Placement, Greedy First-Fit và Greedy Share-All, sau đó phân tích kết quả theo các nhóm chỉ số về khả năng phục vụ, chất lượng dịch vụ, hiệu quả sử dụng tài nguyên và chi phí thực thi. Đây là chương có vai trò kiểm chứng thực nghiệm cho các nội dung đã được đề xuất ở Chương 3.

Như vậy, toàn bộ luận văn được phát triển theo một cấu trúc nhất quán. Chương 1 đặt vấn đề và xác định hướng nghiên cứu. Chương 2 xây dựng nền tảng lý thuyết và chỉ ra khoảng trống nghiên cứu. Chương 3 cụ thể hóa khoảng trống đó thành một phương pháp đề xuất. Chương 4 kiểm chứng phương pháp bằng mô phỏng và phân tích định lượng. Cách tổ chức này giúp luận văn phát triển theo mạch logic từ tổng quan đến đề xuất và từ đề xuất đến đánh giá, bảo đảm tính liên mạch giữa các chương cũng như tính thống nhất của toàn bộ công trình nghiên cứu.

CHƯƠNG 2. NGHIÊN CỨU CÁC PHƯƠNG PHÁP CẢI TIẾN HIỆU NĂNG NFV

2.1. Giới thiệu chương

Sau khi đã trình bày bối cảnh nghiên cứu, mục tiêu và định hướng tiếp cận của luận văn ở Chương 1, chương này tập trung khảo sát cơ sở lý thuyết và các hướng nghiên cứu liên quan trực tiếp đến bài toán cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization). Nếu Chương 1 đặt ra câu hỏi vì sao bài toán này đáng được nghiên cứu, thì Chương 2 có nhiệm vụ trả lời câu hỏi cộng đồng nghiên cứu đã tiếp cận vấn đề này như thế nào, các nhóm giải pháp chính hiện nay là gì, và đâu là khoảng trống có thể được khai thác để xây dựng phương pháp đề xuất ở Chương 3 [1, 3, 7, 8].

Trong những năm gần đây, sự phát triển mạnh mẽ của các dịch vụ số, điện toán đám mây, mạng 5G và các hệ thống phân tán quy mô lớn đã làm cho mô hình mạng dựa trên thiết bị phần cứng chuyên dụng bộc lộ nhiều hạn chế. Các chức năng mạng truyền thống như *tường lửa* (firewall), *cân bằng tải* (load balancer), *phát hiện xâm nhập* (intrusion detection system) hay *dịch địa chỉ mạng* (network address translation) vốn thường được triển khai trên các middlebox độc lập. Cách tiếp cận này gây ra chi phí đầu tư lớn, làm giảm tính linh hoạt khi nâng cấp dịch vụ, đồng thời khiến việc vận hành, mở rộng và tối ưu tài nguyên trở nên phức tạp. Trong bối cảnh đó, *ảo hóa chức năng mạng* (Network Function Virtualization) và *mạng điều khiển bằng phần mềm* (Software-Defined Networking) được xem là hai công nghệ nền tảng giúp tái cấu trúc cách triển khai và quản trị mạng hiện đại [1–3, 5].

Từ góc độ nghiên cứu, NFV không chỉ đơn thuần là việc chuyển các chức năng mạng từ phần cứng sang phần mềm. Khi các *chức năng mạng ảo* (Virtual Network Function) được triển khai trên hạ tầng dùng chung, toàn bộ bài toán vận hành mạng chuyển thành bài toán phân bổ tài nguyên, điều phối dịch vụ, kiểm soát hiệu năng, bảo đảm độ trễ, khả năng mở rộng và duy trì chất lượng phục vụ. Điều này đặc biệt rõ khi các VNF không tồn tại đơn lẻ mà được liên kết thành các *chuỗi chức năng dịch vụ* (Service Function Chain), nơi lưu lượng phải đi qua nhiều chức năng theo đúng thứ tự xử lý. Vì vậy, những vấn đề như *đặt VNF* (VNF placement), *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), *điều hướng lưu lượng* (traffic steering), *phân bổ*

tài nguyên (resource allocation) và *đảm bảo SLA* (SLA guarantees) trở thành những chủ đề trung tâm trong nghiên cứu NFV/SFC [3, 7, 8].

Trong phạm vi luận văn này, Chương 2 không nhằm tổng hợp toàn bộ lĩnh vực NFV, mà được tổ chức theo hướng có chọn lọc, phục vụ trực tiếp cho việc xây dựng phương pháp đề xuất ở Chương 3. Do đó, chương sẽ tập trung vào bốn lớp nội dung chính. Trước hết, luận văn trình bày tổng quan về các công nghệ NFV, SDN và SFC nhằm thiết lập nền tảng khái niệm và mối quan hệ giữa chúng. Tiếp theo, chương đi sâu hơn vào kiến trúc và các thành phần chính của hệ thống NFV, qua đó làm rõ vai trò của hạ tầng, điều phối và các thực thể dịch vụ trong môi trường ảo hóa mạng. Sau đó, các yêu cầu kỹ thuật và các chỉ tiêu hiệu năng quan trọng trong môi trường NFV/SFC sẽ được phân tích nhằm làm rõ vì sao placement là một bài toán nhiều mục tiêu và nhiều ràng buộc. Trên nền tảng đó, luận văn khảo sát các nhóm phương pháp cải tiến hiệu năng đã được nghiên cứu trong tài liệu, đặc biệt là các hướng liên quan trực tiếp đến placement/chaining, workload động, cơ chế dùng chung VNF và các heuristic tối ưu hiệu năng [1, 3, 10, 11].

Cách tổ chức như vậy có ý nghĩa quan trọng đối với toàn bộ cấu trúc luận văn. Một mặt, nó giúp Chương 2 làm tốt vai trò tổng quan học thuật, bảo đảm người đọc có đủ bối cảnh để theo dõi phần phương pháp ở chương sau. Mặt khác, việc khảo sát theo hướng có chọn lọc còn giúp tránh tình trạng Chương 2 bị tách rời khỏi phần đóng góp thực tế của luận văn. Nói cách khác, mọi nội dung được trình bày trong chương này đều hướng đến việc chuẩn bị cơ sở cho Chương 3, nơi luận văn đề xuất phương pháp DWSA-VNF như một heuristic placement có xét đồng thời *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing) [11–13, 15].

Nội dung còn lại của chương được tổ chức như sau. Mục 2.2 trình bày tổng quan về công nghệ NFV, SDN và SFC. Mục 2.3 phân tích kiến trúc và các thành phần chính của hệ thống NFV. Mục 2.4 thảo luận các yêu cầu kỹ thuật và chỉ tiêu hiệu năng trong môi trường NFV/SFC. Mục 2.5 khảo sát các nhóm phương pháp cải tiến hiệu năng NFV trong tài liệu nghiên cứu. Mục 2.6 phân tích sâu các hướng liên quan trực tiếp đến đề tài và chỉ ra khoảng trống nghiên cứu. Cuối cùng, Mục 2.7 đưa ra kết luận chương và làm cầu nối sang Chương 3.

2.2. Tổng quan về công nghệ NFV, SDN và SFC

Sự hình thành và phát triển của các công nghệ *ảo hóa chức năng mạng* (Network Function Virtualization), *mạng điều khiển bằng phần mềm* (Software-Defined Networking) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining) gắn liền với xu hướng mềm hóa hạ tầng mạng trong nhiều năm gần đây. Thay vì tiếp tục phụ thuộc vào mô hình thiết bị mạng chuyên dụng, đắt tiền và khó nâng cấp, cộng đồng nghiên cứu và công nghiệp viễn thông đã chuyển dần sang cách tiếp cận sử dụng phần mềm để triển khai, điều phối và tối ưu dịch vụ mạng. Trong bối cảnh đó, NFV, SDN và SFC không nên được xem là ba công nghệ tách rời, mà là ba thành phần có mối quan hệ bổ sung chặt chẽ với nhau trong việc hiện thực hóa các mạng thế hệ mới [1–3, 6, 17].

2.2.1. Khái niệm và đặc trưng của NFV

Ảo hóa chức năng mạng (Network Function Virtualization) là một cách tiếp cận trong đó các chức năng mạng truyền thống vốn được triển khai trên phần cứng chuyên dụng được chuyển đổi thành các thực thể phần mềm chạy trên hạ tầng tính toán phổ dụng, thường là các máy chủ *thương mại sẵn có* (Commodity Off-The-Shelf hardware). Theo hướng tiếp cận này, các chức năng như firewall, load balancer, NAT, IDS hay DPI không còn buộc phải tồn tại dưới dạng thiết bị vật lý riêng lẻ, mà được cài đặt thành các *chức năng mạng ảo* (Virtual Network Function) có thể khởi tạo, mở rộng, dừng hoặc di chuyển linh hoạt tùy theo nhu cầu của dịch vụ [1, 2, 7].

Ý nghĩa của NFV không chỉ nằm ở việc “ảo hóa” một chức năng mạng, mà quan trọng hơn là ở việc thay đổi toàn bộ triết lý triển khai dịch vụ. Trong mô hình truyền thống, mỗi chức năng mạng chuyên dụng thường gắn với một chu kỳ đầu tư, vận hành và nâng cấp khá cứng nhắc. Khi số lượng dịch vụ gia tăng hoặc khi cần mở rộng quy mô, nhà cung cấp phải mua thêm thiết bị, cấu hình lại hạ tầng và chấp nhận mức độ phức tạp cao trong vận hành. Trái lại, NFV cho phép triển khai dịch vụ mạng bằng phần mềm, từ đó làm giảm *chi phí đầu tư* (Capital Expenditure) và *chi phí vận hành* (Operational Expenditure), đồng thời rút ngắn quá trình đưa dịch vụ mới ra thị trường [1–3, 18]. Đây cũng là lý do vì sao NFV được xem là một trong những công nghệ then chốt của quá trình chuyển đổi số trong viễn thông và điện toán mạng.

Một đặc điểm quan trọng của NFV là sự tách rời giữa chức năng mạng và nền phần cứng thực thi. Khi các VNF được cài đặt dưới dạng phần mềm, chúng có thể được điều phối linh hoạt trên các hạ tầng khác nhau như trung tâm dữ liệu, điện toán biên hoặc các môi trường đám mây phân tán. Tuy nhiên, chính sự linh hoạt này cũng

dẫn đến những thách thức mới. Một VNF chỉ thực sự phát huy giá trị nếu nó được đặt lên hạ tầng phù hợp, có đủ tài nguyên tính toán, đủ tài nguyên mạng, và được kết nối đúng cách với các VNF khác trong cùng một chuỗi dịch vụ. Điều này làm xuất hiện hàng loạt bài toán như placement, chaining, scaling, monitoring và resource allocation, vốn trở thành trung tâm của các nghiên cứu hiện nay [3, 7–9].

Một khía cạnh khác cần lưu ý là NFV không tồn tại đơn lẻ như một công nghệ ảo hóa thuần túy. Trên thực tế, để một môi trường NFV vận hành hiệu quả, cần có các thành phần quản lý và điều phối, cơ chế giám sát tài nguyên, các phương tiện định tuyến lưu lượng, cũng như các nền tảng hỗ trợ phát triển và triển khai VNF. Paper về NFV platforms cho thấy không gian thiết kế của một nền tảng NFV là rất rộng, trải từ các API phát triển VNF, mô hình thực thi, quản lý trạng thái, cho đến các kỹ thuật tăng tốc datapath như zero-copy, batching, computation offloading và packet I/O frameworks. Điều này cho thấy rằng NFV không chỉ là một công nghệ hạ tầng, mà còn là một hệ sinh thái kỹ thuật có chiều sâu đáng kể [1, 18, 19].

2.2.2. *Khái niệm và vai trò của SDN*

Nếu NFV làm thay đổi cách triển khai các chức năng mạng, thì *mạng điều khiển bằng phần mềm* (Software-Defined Networking) làm thay đổi cách điều khiển và quản trị mạng. Ý tưởng cốt lõi của SDN là tách biệt *mặt phẳng điều khiển* (control plane) khỏi *mặt phẳng dữ liệu* (data plane), từ đó cho phép logic điều khiển được tập trung hóa và lập trình hóa thông qua một thực thể điều khiển có cái nhìn toàn cục hơn về trạng thái mạng. Với cách tiếp cận này, các thiết bị chuyển mạch và định tuyến không còn phải tự mang toàn bộ logic điều khiển phức tạp, mà có thể được lập trình hoặc cấu hình linh hoạt bởi bộ điều khiển SDN [1, 3, 4].

Vai trò của SDN trong bối cảnh NFV là đặc biệt quan trọng. Khi nhiều VNF được triển khai linh hoạt trên hạ tầng phân tán, lưu lượng mạng phải được dẫn đi đúng chuỗi xử lý dịch vụ tương ứng. Điều này đòi hỏi một cơ chế có khả năng thiết lập và thay đổi luồng lưu lượng một cách mềm dẻo, theo đúng cấu trúc dịch vụ và theo đúng trạng thái hiện tại của hạ tầng. SDN cung cấp chính xác năng lực đó thông qua các cơ chế như điều khiển tập trung, lập trình đường đi và triển khai chính sách forwarding linh hoạt. Vì vậy, trong nhiều hệ thống hiện đại, NFV và SDN được xem là hai công nghệ bổ trợ lẫn nhau: NFV cung cấp các chức năng mạng dưới dạng phần mềm, còn SDN cung cấp khả năng điều hướng và kiểm soát lưu lượng để các chức năng đó có thể phối hợp hiệu quả [3, 11, 17, 20].

Từ góc độ dịch vụ, SDN còn giúp tăng mức tự động hóa và khả năng thích ứng của hệ thống. Khi lưu lượng thay đổi, khi một VNF được scale-out, hoặc khi topology logic của dịch vụ cần điều chỉnh, SDN cho phép phản ứng nhanh hơn so với các mô hình cấu hình thủ công truyền thống. Chính nhờ đặc điểm này mà SDN thường xuất hiện như một lớp điều phối mạng rất quan trọng trong các kiến trúc NFV/SFC hiện đại. Nói cách khác, nếu NFV làm cho chức năng mạng trở nên mềm dẻo hơn, thì SDN làm cho quá trình liên kết và vận hành các chức năng đó cũng trở nên mềm dẻo hơn [1, 3, 4].

2.2.3. *Khái niệm và vai trò của SFC*

Trong các hệ thống mạng hiện đại, một yêu cầu dịch vụ hiếm khi chỉ cần đi qua duy nhất một chức năng mạng. Trên thực tế, một luồng dữ liệu thường phải đi qua nhiều chức năng theo một thứ tự xác định, chẳng hạn đi qua firewall trước, rồi đến IDS, sau đó đến load balancer hoặc NAT. Chính nhu cầu này dẫn đến khái niệm *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), trong đó các VNF được liên kết lại thành một chuỗi logic để xử lý lưu lượng theo đúng chính sách dịch vụ [3, 5, 6].

SFC là một khái niệm đặc biệt quan trọng vì nó biến bài toán “triển khai từng VNF riêng lẻ” thành bài toán “triển khai và điều phối cả một chuỗi chức năng”. Khi đó, mỗi quyết định đặt một VNF lên node nào không còn là một quyết định độc lập, mà ảnh hưởng trực tiếp đến đường đi của lưu lượng, đến độ trễ tích lũy, đến mức sử dụng băng thông và đến khả năng placement của các VNF tiếp theo trong cùng một chuỗi. Vì vậy, placement trong môi trường SFC luôn có độ phức tạp cao hơn placement của một chức năng đơn lẻ, và đây cũng là lý do vì sao SFC thường gắn liền với các bài toán tối ưu và heuristic về mapping, embedding, resource allocation và traffic steering [3, 7, 8, 10].

Một điểm đáng chú ý là SFC không chỉ là một chuỗi tuyến tính cố định trên lý thuyết, mà là một cơ chế mô tả logic dịch vụ trong môi trường có thể thay đổi động. Các yêu cầu người dùng khác nhau có thể đòi hỏi các chuỗi khác nhau, với số lượng VNF khác nhau, thứ tự khác nhau và ngưỡng *chất lượng dịch vụ* (Quality of Service) khác nhau. Vì vậy, SFC làm nổi bật nhu cầu cần có một phương pháp placement đủ linh hoạt để thích nghi với sự đa dạng và biến động của dịch vụ. Đây cũng là điểm mà luận văn sẽ khai thác trong các chương sau, khi chuyển từ tổng quan công nghệ sang bài toán cải tiến hiệu năng placement/chaining [3, 5, 21].

2.2.4. *Mối quan hệ giữa NFV, SDN và SFC*

Mặc dù NFV, SDN và SFC có thể được nghiên cứu như các chủ đề riêng, nhưng trong thực tế triển khai mạng hiện đại, ba công nghệ này thường vận hành trong một quan hệ bổ sung rất chặt chẽ. NFV cung cấp các VNF như những khối chức năng mạng có thể triển khai bằng phần mềm. SFC xác định thứ tự logic mà các khối chức năng đó phải tham gia để tạo thành một dịch vụ hoàn chỉnh. SDN cung cấp cơ chế điều khiển và lập trình lưu lượng để dữ liệu thực sự được đưa đi qua đúng chuỗi chức năng đã xác định. Khi kết hợp lại, ba công nghệ này tạo thành một khung mềm hóa mạng rất mạnh, cho phép nhà cung cấp dịch vụ vừa linh hoạt trong thiết kế dịch vụ, vừa chủ động trong khai thác tài nguyên, vừa có khả năng tự động hóa vận hành ở mức cao [1, 3, 6, 17].

Chính mối quan hệ này là nền tảng cho bài toán mà luận văn đang xem xét. Nếu chỉ có NFV mà không có một cơ chế chain và điều khiển lưu lượng, thì các VNF chỉ tồn tại như những khối xử lý rời rạc. Nếu chỉ có SDN mà không có NFV, mạng có thể lập trình được nhưng vẫn bị giới hạn bởi các chức năng phần cứng truyền thống. Nếu có NFV và SDN nhưng thiếu logic SFC, việc cung cấp dịch vụ nhiều chức năng cho người dùng cuối sẽ thiếu nhất quán. Vì vậy, khi nói đến cải tiến hiệu năng trong môi trường NFV, thực chất cần nhìn vấn đề trong tam giác NFV–SDN–SFC, nơi placement, chaining và traffic steering cùng tham gia quyết định chất lượng cuối cùng của dịch vụ [1, 3, 8, 20].

Một điểm cần nhấn mạnh thêm là sự kết hợp giữa NFV, SDN và SFC không chỉ mang ý nghĩa công nghệ, mà còn làm thay đổi bản chất của nhiều bài toán tối ưu trong mạng hiện đại. Trong mô hình truyền thống, các quyết định về vị trí xử lý, đường đi của lưu lượng và khả năng mở rộng dịch vụ thường gắn chặt với cấu hình phần cứng và ít biến động trong thời gian ngắn. Trái lại, trong môi trường NFV–SDN–SFC, các quyết định đó trở thành các biến số có thể thay đổi động theo trạng thái tài nguyên, trạng thái lưu lượng và yêu cầu dịch vụ. Điều này làm cho các bài toán như *đặt VNF* (VNF placement), *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), *điều hướng lưu lượng* (traffic steering) và *điều phối dịch vụ* (service orchestration) trở nên quan trọng hơn nhiều so với trước đây [3, 17, 20].

Từ góc nhìn nghiên cứu, chính sự thay đổi này giải thích vì sao cộng đồng học thuật không chỉ quan tâm đến việc ảo hóa chức năng mạng, mà còn quan tâm đến cách tối ưu vị trí triển khai, cách tổ chức chuỗi xử lý và cách sử dụng tài nguyên chung sao

cho hiệu quả. Nói cách khác, NFV, SDN và SFC không chỉ là ba công nghệ nền, mà còn là ba nguồn tạo ra các ràng buộc và cơ hội tối ưu mới trong môi trường mạng mềm hóa. Đây cũng là điểm làm cho bài toán của luận văn trở nên có ý nghĩa: cải tiến hiệu năng trong NFV thực chất phải được nhìn trong mối liên hệ tổng thể giữa hạ tầng ảo hóa, điều khiển lưu lượng và cấu trúc dịch vụ.

Từ các phân tích trên có thể thấy rằng NFV, SDN và SFC tạo thành nền công nghệ trực tiếp của bài toán placement/chaining trong luận văn. NFV làm xuất hiện các VNF cần được đặt lên hạ tầng, SFC tạo ra ràng buộc thứ tự xử lý dịch vụ, còn SDN cung cấp cơ chế hỗ trợ điều hướng lưu lượng và điều phối dịch vụ trong mạng mềm hóa. Trên nền tảng này, các vấn đề sâu hơn như kiến trúc hệ thống NFV, vai trò của MANO, NFVI, resource allocation và các chỉ tiêu hiệu năng sẽ được trình bày ở các mục tiếp theo của chương [1, 3, 7].

2.3. Kiến trúc và các thành phần chính của hệ thống NFV

Để nghiên cứu các phương pháp cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization), trước hết cần làm rõ kiến trúc tổng thể của một hệ thống NFV và vai trò của các thành phần chính trong kiến trúc đó. Về bản chất, NFV không chỉ là việc chạy một số *chức năng mạng ảo* (Virtual Network Function) trên máy chủ phổ dụng, mà là một hệ thống dịch vụ hoàn chỉnh bao gồm hạ tầng thực thi, các thực thể quản lý và điều phối, cũng như các thành phần chức năng phục vụ chuỗi dịch vụ. Chính cấu trúc kiến trúc này quyết định cách tài nguyên được cấp phát, cách dịch vụ được triển khai, cách các VNF được liên kết với nhau và cách toàn bộ hệ thống được giám sát, mở rộng hoặc phục hồi khi có biến động [1–3].

Theo các tài liệu tổng quan hiện nay, mặc dù nhiều nền tảng NFV cụ thể có thể khác nhau ở mức hiện thực, một kiến trúc NFV điển hình vẫn thường xoay quanh ba khối chính: *mặt phẳng quản lý và điều phối NFV* (NFV Management and Orchestration plane), *mặt phẳng dịch vụ* (service plane) và *hạ tầng NFV* (NFV Infrastructure). Mặt phẳng dịch vụ chứa các VNF và các *chuỗi chức năng dịch vụ* (Service Function Chain) tạo thành các dịch vụ mạng cụ thể. Hạ tầng NFV cung cấp tài nguyên tính toán, lưu trữ và mạng để các VNF có thể vận hành. Trong khi đó, mặt phẳng quản lý và điều phối giữ vai trò trung tâm trong việc triển khai, giám sát, phân bổ tài nguyên, mở rộng hoặc tái cấu hình dịch vụ theo trạng thái thực tế của hệ thống [1, 3, 19].

2.3.1. Kiến trúc tổng quát của hệ thống NFV

Ở mức khái quát, kiến trúc NFV có thể được hiểu như một cấu trúc nhiều lớp. Ở lớp thấp nhất là *hạ tầng vật lý và ảo hóa* (NFV Infrastructure), bao gồm các máy chủ, thiết bị mạng, bộ nhớ, tài nguyên lưu trữ và các lớp ảo hóa cần thiết để chạy các VNF. Ở lớp trung gian là các VNF cùng với các chuỗi dịch vụ mà chúng tạo thành. Ở lớp trên cùng là các thực thể quản lý và điều phối chịu trách nhiệm giám sát, lập kế hoạch triển khai, cấp phát tài nguyên, scale, migrate hoặc phục hồi dịch vụ. Cách tổ chức theo nhiều lớp như vậy cho phép tách biệt bài toán hạ tầng, bài toán dịch vụ và bài toán điều phối, nhưng đồng thời cũng tạo ra những quan hệ phụ thuộc rất chặt giữa các lớp này [1, 17].

Trong một hệ thống NFV hiện đại, chuỗi cung cấp dịch vụ không còn diễn ra theo kiểu thiết bị nào xử lý phần việc đó một cách cố định như trong mạng truyền thống. Thay vào đó, dịch vụ được mô tả dưới dạng chuỗi logic, được triển khai động trên hạ tầng thông qua các quyết định placement và chaining. Vì vậy, kiến trúc tổng thể của NFV phải hỗ trợ được ít nhất ba yêu cầu. Thứ nhất, hạ tầng phải đủ linh hoạt để chứa nhiều loại VNF khác nhau. Thứ hai, cơ chế quản lý phải có khả năng điều phối và giám sát tài nguyên theo thời gian thực hoặc gần thời gian thực. Thứ ba, luồng lưu lượng phải được dẫn đi đúng theo cấu trúc dịch vụ đã xác định. Những yêu cầu này lý giải vì sao NFV thường gắn chặt với SDN, MANO và các kỹ thuật tự động hóa vận hành [1, 3, 22].

Từ góc nhìn của luận văn, việc nắm vững kiến trúc tổng quát của hệ thống NFV có ý nghĩa rất quan trọng. Bài toán placement được đề xuất ở Chương 3 không thể được hiểu đơn thuần như bài toán gán một VNF lên một node, mà phải được nhìn như một quyết định xảy ra trong một kiến trúc có nhiều thành phần liên hệ với nhau. Cụ thể, placement chịu ràng buộc bởi tài nguyên của NFVI, chịu điều phối bởi MANO, và chịu tác động bởi cấu trúc chuỗi dịch vụ ở service plane. Do đó, trước khi khảo sát các phương pháp tối ưu placement hay resource allocation, cần làm rõ vai trò của từng thành phần trong kiến trúc này.

2.3.2. Hạ tầng NFV và vai trò của NFVI

Hạ tầng NFV (NFV Infrastructure, NFVI) là lớp nền tảng vật lý và ảo hóa cho toàn bộ môi trường NFV. NFVI bao gồm các tài nguyên tính toán, lưu trữ và mạng phân bố trên nhiều nút hạ tầng khác nhau. Trong nhiều tài liệu, mỗi nút trong NFVI có thể là máy chủ COTS, switch, router, SmartNIC hoặc các thiết bị có khả năng lập

trình và hỗ trợ chạy VNF ở các mức độ khác nhau. Cốt lõi của NFVI là cung cấp môi trường để các VNF có thể được khởi tạo, thực thi, liên kết và mở rộng theo nhu cầu dịch vụ [1, 2].

Một đặc trưng rất quan trọng của NFVI là tính dị thể. Không phải mọi nút hạ tầng đều có cùng cấu hình phần cứng, cùng loại tài nguyên hoặc cùng khả năng hỗ trợ các kiểu ảo hóa. Một số nền tảng ưu tiên triển khai VNF trên máy ảo, trong khi một số khác ưu tiên container hoặc thậm chí tiến trình mức người dùng để giảm overhead. Điều này tạo ra không gian thiết kế phong phú, nhưng đồng thời cũng làm placement trở nên phức tạp hơn vì không phải mọi VNF đều có thể đặt ở mọi node theo cùng một cách [1, 18].

Từ góc nhìn hiệu năng, NFVI không chỉ đơn thuần là tập các node có *bộ xử lý trung tâm* (CPU) và bộ nhớ. Các lựa chọn thiết kế ở lớp hạ tầng, chẳng hạn kỹ thuật ảo hóa, cơ chế packet I/O, hình thức interconnect giữa các VNF và khả năng tăng tốc datapath, đều tác động trực tiếp đến hiệu năng dịch vụ. Chẳng hạn, sử dụng máy ảo có thể cho mức cô lập tốt hơn nhưng thường mang theo overhead lớn hơn; container có thể nhẹ hơn nhưng yếu hơn về cô lập; tiến trình người dùng có thể đạt hiệu năng cao hơn nhưng đánh đổi về bảo mật và quản lý tài nguyên. Vì vậy, trong nghiên cứu NFV, hiệu năng của một phương pháp placement không thể tách rời khỏi bối cảnh của NFVI mà nó hoạt động trên [1, 23].

Một ý nghĩa quan trọng khác của NFVI đối với luận văn là đây chính là “không gian placement” của các VNF. Mọi thuật toán placement cuối cùng đều phải quyết định chức năng nào được đặt trên node nào của NFVI, sử dụng bao nhiêu tài nguyên, với liên kết mạng nào và theo ràng buộc hiệu năng nào. Do đó, việc trình bày NFVI trong Chương 2 sẽ tạo nền trực tiếp cho mô hình hệ thống ở Chương 3, nơi luận văn biểu diễn hạ tầng bằng đồ thị gồm các node và liên kết hữu hạn tài nguyên.

2.3.3. *Mặt phẳng quản lý và điều phối NFV*

Trong các kiến trúc chuẩn hóa của NFV, *mặt phẳng quản lý và điều phối NFV* (NFV Management and Orchestration, MANO) được xem là thành phần trung tâm điều hành toàn bộ vòng đời của dịch vụ. MANO thường bao gồm ba thực thể chính là *bộ điều phối NFV* (NFV Orchestrator, NFVO), *bộ quản lý hạ tầng ảo hóa* (Virtualized Infrastructure Manager, VIM) và *bộ quản lý chức năng mạng ảo* (VNF Manager, VNFM). Ba thành phần này không hoạt động tách biệt, mà phối hợp chặt chẽ với nhau để bảo đảm rằng dịch vụ có thể được triển khai, mở rộng, giám sát và kết thúc

đúng theo yêu cầu vận hành [1, 19].

Bộ điều phối NFV (NFVO) giữ vai trò ở mức cao nhất trong quá trình triển khai dịch vụ. NFVO tiếp nhận yêu cầu dịch vụ, xử lý mô tả dịch vụ, tính toán placement, quyết định cách xâu chuỗi các VNF, đồng thời phối hợp với các thành phần còn lại để hiện thực hóa quyết định đó trên hạ tầng. Trong nhiều nền tảng, NFVO không chỉ làm placement ở thời điểm đầu mà còn tiếp tục theo dõi và điều chỉnh placement khi tải hệ thống thay đổi hoặc khi xuất hiện nhu cầu scale-in, scale-out hay failover. Đây là điểm đặc biệt quan trọng đối với bài toán của luận văn, bởi placement trong môi trường NFV không phải là một quyết định tĩnh một lần, mà có thể cần được điều chỉnh theo trạng thái vận hành [1, 22].

Bộ quản lý hạ tầng ảo hóa (VIM) chịu trách nhiệm ở lớp tài nguyên hạ tầng. VIM thực hiện việc provision, release hoặc upgrade tài nguyên của NFVI, đồng thời quản lý ánh xạ giữa tài nguyên ảo và tài nguyên vật lý. Ngoài ra, VIM còn tham gia cấu hình các giao diện ảo, logical links và dữ liệu trạng thái tài nguyên cho các thực thể điều phối khác. Có thể nói, nếu NFVO quyết định “dịch vụ nên đi đâu”, thì VIM là thành phần bảo đảm “hạ tầng có thể cung cấp tài nguyên để dịch vụ thực sự được đặt ở đó” [1, 3].

Bộ quản lý chức năng mạng ảo (VNFM) chịu trách nhiệm trực tiếp với vòng đời của các VNF. Các thao tác như instantiate, scale, upgrade, terminate hoặc update trạng thái của từng VNF thường được VNFM đảm nhiệm. Trong môi trường có nhiều loại VNF khác nhau, VNFM đóng vai trò như một lớp trung gian giúp dịch vụ logic ở tầng trên được chuyển hóa thành hành động thực thi cụ thể ở tầng dưới. Điều này đặc biệt quan trọng khi hệ thống cần phản ứng động trước sự thay đổi của lưu lượng, lỗi hạ tầng hoặc yêu cầu mới từ phía dịch vụ [1, 19].

Một điểm đáng chú ý là MANO không chỉ là một khối điều phối hành chính, mà trực tiếp liên quan tới hiệu năng của hệ thống. Placement, scheduling, monitoring và scaling đều là các thao tác do MANO điều phối, và chất lượng của các thao tác này quyết định liệu hệ thống có sử dụng tài nguyên hiệu quả hay không, có duy trì được delay hay không, và có bảo đảm SLA hay không. Vì vậy, nhiều tài liệu xem resource allocation, orchestration và placement như các chủ đề không thể tách rời trong NFV [7, 9, 22].

Từ phương diện kiến trúc, có thể xem MANO là lớp biến các quyết định logic của dịch vụ thành các thao tác vận hành cụ thể trên hạ tầng. Điều này đặc biệt quan

trọng đối với bài toán placement/chaining, vì placement không chỉ là một quyết định mang tính thuật toán, mà còn là một quyết định phải được thực thi thông qua chuỗi thao tác quản lý, cấp phát tài nguyên, kích hoạt VNF, thiết lập liên kết logic và cập nhật trạng thái của hệ thống. Nếu thiếu lớp MANO hoặc nếu MANO không đủ khả năng phản ứng trước các thay đổi của hạ tầng và dịch vụ, ngay cả một thuật toán placement tốt về mặt lý thuyết cũng rất khó phát huy hiệu quả trong thực tế [1, 19, 22].

Chính vì vậy, trong nhiều nghiên cứu hiện nay, placement không được xem như một bài toán độc lập nằm ngoài kiến trúc hệ thống, mà được xem là một thành phần của orchestration tổng thể. Cách nhìn này có ý nghĩa quan trọng đối với luận văn, bởi phương pháp đề xuất ở Chương 3 tuy được xây dựng ở mức heuristic placement, nhưng vẫn phải nhất quán với logic chung của môi trường NFV, nơi mọi quyết định placement cuối cùng đều phải quy về khả năng thực thi trên hạ tầng và khả năng điều phối của hệ thống quản lý.

2.3.4. Mặt phẳng dịch vụ và vòng đời VNF

Nếu NFVI là lớp hạ tầng và MANO là lớp điều phối, thì *mặt phẳng dịch vụ* (service plane) là nơi các dịch vụ mạng thực sự được hiện thực hóa thông qua các VNF và các chuỗi chức năng dịch vụ. Ở lớp này, mỗi dịch vụ được mô tả như một cấu trúc gồm nhiều chức năng liên tiếp, phối hợp với nhau để cung cấp một đầu ra mạng mong muốn. Tính đa dạng của service plane là rất lớn, bởi các loại VNF có thể khác nhau về đặc tính xử lý, mức tiêu thụ tài nguyên, khả năng mở rộng, khả năng dùng chung và mức độ nhạy cảm với trạng thái nội bộ [1, 3].

Một nền tảng NFV điển hình thường phải hỗ trợ toàn bộ vòng đời của VNF, từ xây dựng, triển khai, thực thi, giám sát cho tới scale, migrate và terminate. Các survey về NFV platforms cho thấy nhiều platform được phân loại dựa trên chính vòng đời này, vì mỗi giai đoạn đều liên quan đến những quyết định kỹ thuật khác nhau. Ở giai đoạn phát triển, platform cần hỗ trợ API hoặc ngôn ngữ phù hợp cho nhà phát triển VNF. Ở giai đoạn placement và deployment, platform cần hỗ trợ mô tả dịch vụ, xử lý graph, pre-processing và tối ưu placement. Ở giai đoạn runtime, platform cần có công cụ giám sát, scheduling, state coordination và secure execution [1].

Đối với luận văn, điều quan trọng nhất ở service plane là nhận thức rằng placement không thể chỉ được xem như bài toán ở lớp hạ tầng. Placement luôn gắn với đặc tính dịch vụ. Một VNF stateful như firewall hoặc IDS có thể có yêu cầu khác về sharing và placement so với NAT hoặc load balancer. Một chuỗi dịch vụ dài có thể

nhảy hơn với delay tích lũy so với một chuỗi ngắn. Một số dịch vụ có thể chịu tải biến thiên mạnh theo thời gian, trong khi một số khác có profile ổn định hơn. Do đó, service plane là nơi các ràng buộc logic của dịch vụ được đưa vào bài toán kỹ thuật của placement/chaining.

2.3.5. Một số lựa chọn thiết kế nền tảng và ảnh hưởng đến hiệu năng

Một lý do khiến NFV trở thành một lĩnh vực nghiên cứu rất rộng là vì tồn tại nhiều *lựa chọn thiết kế* (design choices) khác nhau ở nhiều lớp của hệ thống. Theo các khảo sát về NFV platforms, những lựa chọn này có thể xuất hiện từ mức API phát triển VNF, mô hình thực thi, quản lý trạng thái, kỹ thuật packet I/O, cho đến công nghệ ảo hóa và các cơ chế tăng tốc datapath [1]. Điều quan trọng là không có một lựa chọn nào là tốt nhất trong mọi trường hợp; mỗi lựa chọn thường kéo theo một sự đánh đổi giữa hiệu năng, độ linh hoạt, mức độ cô lập và chi phí vận hành.

Một trong các lựa chọn quan trọng nhất là *công nghệ ảo hóa* (virtualization technique). Máy ảo thường cho độ cô lập tốt nhưng mang theo overhead lớn hơn về bộ nhớ và chi phí migrate. Container nhẹ hơn, khởi động nhanh hơn và có mật độ triển khai cao hơn, nhưng thường yếu hơn về mặt cô lập. Triển khai VNF dưới dạng process có thể tăng hiệu năng hơn nữa nhưng làm nảy sinh các thách thức về quản lý tài nguyên và bảo mật. Chính sự đa dạng này khiến các nền tảng NFV khác nhau có thể tối ưu theo những hướng rất khác nhau [1, 18].

Bên cạnh ảo hóa, *kỹ thuật packet I/O* cũng là một nhân tố quan trọng đối với hiệu năng. Các giải pháp kernel-bypassing như DPDK hoặc netmap thường được sử dụng để vượt qua overhead của kernel network stack truyền thống và đạt được tốc độ xử lý cao hơn. Ngoài ra, các kỹ thuật như zero-copy, batching, pre-allocation, cache optimization, parallel execution và offloading cũng thường được triển khai để tăng hiệu năng xử lý gói tin [1, 23]. Những yếu tố này cho thấy rằng hiệu năng trong NFV không chỉ là vấn đề placement ở cấp logic dịch vụ, mà còn bị ảnh hưởng mạnh bởi lựa chọn hiện thực của nền tảng.

Một khía cạnh khác là *quản lý trạng thái* (state management). Khi số lượng VNF stateful tăng lên, việc scale, migrate hoặc failover mà vẫn giữ đúng xử lý lưu lượng trở nên rất khó. Đây là lý do vì sao nhiều nghiên cứu tách riêng bài toán state coordination hoặc coi đây là một ràng buộc quan trọng khi tối ưu platform. Mặc dù luận văn này không đi sâu vào bài toán migration hoặc failover, việc nhắc tới state management ở đây là cần thiết để giải thích vì sao cơ chế *chia sẻ có chọn lọc* (selective

sharing) trong Chương 3 chỉ được áp dụng cho một số nhóm VNF nhất định thay vì cho mọi loại VNF.

Ngoài các yếu tố nêu trên, một điểm cũng thường được nhấn mạnh trong các nghiên cứu về nền tảng NFV là mối quan hệ giữa hiệu năng cực đại và tính linh hoạt của hệ thống. Trong nhiều trường hợp, một nền tảng có thể đạt hiệu năng xử lý rất cao nếu được tối ưu hóa mạnh cho một loại VNF, một loại packet I/O hoặc một kiểu ảo hóa cụ thể. Tuy nhiên, mức tối ưu đó thường đi kèm với sự đánh đổi về tính tổng quát, khả năng di động của dịch vụ hoặc mức độ dễ dàng khi orchestration ở quy mô lớn [1, 18]. Điều này cho thấy hiệu năng trong NFV không nên được hiểu đơn giản như tốc độ xử lý gói tin cực đại, mà cần được hiểu theo nghĩa rộng hơn, bao gồm cả khả năng triển khai linh hoạt, khả năng điều phối nhất quán và khả năng duy trì chất lượng dịch vụ trong nhiều kịch bản vận hành khác nhau.

Đối với luận văn này, nhận định trên có ý nghĩa phương pháp luận rõ ràng. Mục tiêu của luận văn không phải là tối ưu hóa ở mức nền tảng packet processing hay tối ưu hóa micro-architecture của VNF, mà là tìm kiếm một cách placement/chaining hợp lý trong một môi trường đã giả định có nền thực thi đủ ổn định. Việc làm rõ ranh giới này giúp luận văn tập trung đúng trọng tâm nghiên cứu, đồng thời tránh mở rộng quá sâu sang các lớp tối ưu không nằm trong phạm vi của đề tài.

Từ những phân tích trên có thể thấy rằng kiến trúc và các thành phần của hệ thống NFV tạo nên bối cảnh trực tiếp cho bài toán placement/chaining. NFVI xác định không gian tài nguyên hữu hạn, MANO điều phối cách sử dụng tài nguyên đó, service plane đưa ra các ràng buộc logic của dịch vụ, còn các lựa chọn thiết kế platform quyết định ngưỡng hiệu năng mà hệ thống có thể đạt được. Trên nền tảng đó, phần tiếp theo của chương sẽ trình bày các yêu cầu kỹ thuật và các chỉ tiêu hiệu năng trong môi trường NFV/SFC, làm cơ sở để đánh giá và so sánh các phương pháp cải tiến hiệu năng.

2.4. Các yêu cầu kỹ thuật và chỉ tiêu hiệu năng trong môi trường NFV/SFC

Khi các chức năng mạng được ảo hóa và triển khai trên hạ tầng dùng chung, hiệu năng của hệ thống không còn được quyết định chủ yếu bởi năng lực của một thiết bị phần cứng riêng lẻ, mà trở thành kết quả của nhiều thành phần phối hợp với nhau. Trong môi trường NFV kết hợp với *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), một dịch vụ chỉ thực sự được cung cấp tốt khi các VNF trong chuỗi được đặt hợp lý, lưu lượng được dẫn đúng đường, tài nguyên được cấp phát đủ và chất lượng

phục vụ vẫn nằm trong ngưỡng cho phép. Điều này làm cho việc xác định các yêu cầu kỹ thuật và các chỉ tiêu hiệu năng trở thành một bước rất quan trọng trước khi khảo sát các phương pháp cải tiến hiệu năng [3, 7, 9].

2.4.1. Yêu cầu về tài nguyên tính toán và tài nguyên mạng

Yêu cầu cơ bản nhất trong môi trường NFV là yêu cầu về tài nguyên tính toán và tài nguyên mạng. Mỗi VNF khi được triển khai đều tiêu tốn một phần *bộ xử lý trung tâm* (CPU), bộ nhớ và đôi khi cả tài nguyên lưu trữ. Đồng thời, khi nhiều VNF được liên kết thành chuỗi dịch vụ, lưu lượng giữa các VNF sẽ tiếp tục tiêu thụ tài nguyên bằng thông trên các liên kết vật lý. Vì vậy, placement trong NFV luôn là một bài toán phân bổ đồng thời trên cả node và link, thay vì chỉ là bài toán chọn máy chủ còn nhiều CPU nhất [7, 8, 24].

Tài nguyên tính toán không chỉ cần xét ở mức tổng dung lượng, mà còn cần xét ở cách tài nguyên được phân bổ giữa các node. Một placement thiếu hợp lý có thể khiến một số node bị quá tải trong khi các node khác vẫn còn dư thừa tài nguyên. Tương tự, tài nguyên mạng cũng không chỉ là tổng băng thông của toàn hệ thống, mà còn là tình trạng các liên kết cụ thể trên đường đi giữa các VNF. Chính vì vậy, nhiều công trình nghiên cứu về placement và resource allocation trong NFV đều nhấn mạnh rằng hiệu quả hệ thống phải được nhìn theo hướng phân bổ tài nguyên, chứ không thể chỉ nhìn theo tổng dung lượng danh nghĩa [7, 9, 21].

2.4.2. Yêu cầu về độ trễ và chất lượng dịch vụ

Bên cạnh tài nguyên, *độ trễ đầu cuối* (end-to-end delay) là một yêu cầu kỹ thuật đặc biệt quan trọng trong NFV/SFC. Một chuỗi dịch vụ có thể được placement thành công về mặt tài nguyên nhưng vẫn không có giá trị thực tiễn nếu tổng độ trễ vượt quá ngưỡng mà dịch vụ cho phép. Trong các môi trường dịch vụ hiện đại, nhiều ứng dụng yêu cầu độ trễ rất chặt, đặc biệt là các dịch vụ thời gian thực, dịch vụ biên hoặc các ứng dụng nhạy với trải nghiệm người dùng. Vì vậy, placement không thể chỉ quan tâm đến việc “có đặt được hay không”, mà còn phải quan tâm đến việc “đặt như thế nào để không làm xấu đi chất lượng dịch vụ” [3, 16, 21].

Trong bối cảnh của SFC, độ trễ không chỉ đến từ bản thân node xử lý, mà còn đến từ đường đi giữa các VNF liên tiếp trong chuỗi. Khi chuỗi dịch vụ càng dài, delay tích lũy càng dễ tăng nhanh. Điều này lý giải vì sao nhiều hướng nghiên cứu gần đây xem delay-aware placement là một lớp riêng trong bài toán tối ưu hiệu năng NFV. Với luận văn này, việc nhấn mạnh yêu cầu về delay ở Chương 2 là rất cần thiết, vì sau này

DWSA-VNF trong Chương 3 cũng lấy delay như một trong các tiêu chí trọng tâm của quá trình sàng lọc và chấm điểm placement.

2.4.3. Yêu cầu về thông lượng và khả năng phục vụ dịch vụ

Nếu delay phản ánh chất lượng phục vụ, thì *thông lượng mạng* (network throughput) và *tỷ lệ chấp nhận yêu cầu* (acceptance ratio) phản ánh năng lực phục vụ của hệ thống. Trong môi trường NFV, một thuật toán placement tốt không chỉ cần giữ độ trễ trong ngưỡng chấp nhận, mà còn phải cho phép hệ thống phục vụ được nhiều lưu lượng hữu ích hơn và chấp nhận được nhiều yêu cầu dịch vụ hơn. Đây là hai chỉ số có ý nghĩa thực tiễn rất rõ ràng, vì chúng phản ánh trực tiếp hiệu quả khai thác hạ tầng [11, 16].

Thông lượng thường được hiểu là tổng lượng dịch vụ hữu ích mà hệ thống phục vụ được trong một khoảng thời gian hoặc một kịch bản nhất định. Trong khi đó, acceptance ratio cho biết có bao nhiêu yêu cầu trong tổng số yêu cầu đầu vào được hệ thống tiếp nhận thành công. Hai chỉ số này liên quan chặt chẽ nhưng không đồng nhất. Một phương pháp có thể chấp nhận nhiều yêu cầu nhỏ nhưng throughput không cao, hoặc ngược lại. Vì vậy, trong các nghiên cứu placement hiện đại, hai chỉ số này thường được sử dụng đồng thời để có cái nhìn đầy đủ hơn về khả năng phục vụ của hệ thống [11, 16, 25].

2.4.4. Yêu cầu về hiệu quả sử dụng tài nguyên

Hiệu năng của hệ thống NFV không chỉ nằm ở số lượng dịch vụ được chấp nhận, mà còn ở cách tài nguyên được sử dụng để đạt được kết quả đó. Một thuật toán placement có thể cho acceptance ratio cao nhưng đổi lại bằng việc kích hoạt quá nhiều node, tạo quá nhiều instance mới hoặc gây ra phân bố tài nguyên rất kém cân bằng. Trong trường hợp đó, kết quả tuy có vẻ tốt ở một chỉ số nhưng lại không bền vững về vận hành. Đây là lý do vì sao *hiệu quả sử dụng tài nguyên* (resource efficiency) được xem là một yêu cầu kỹ thuật cốt lõi trong nhiều khảo sát về NFV [7–9].

Ở mức thực nghiệm, hiệu quả sử dụng tài nguyên thường được phản ánh qua các chỉ số như CPU utilization, memory utilization và bandwidth utilization. Các chỉ số này cho biết mức độ hệ thống tận dụng được tài nguyên sẵn có, cũng như mức độ cân bằng hay mất cân bằng trong khai thác hạ tầng. Nếu tài nguyên được sử dụng quá phân tán hoặc quá dồn cục vào một số điểm, hệ thống dễ gặp nghẽn hoặc phải mở rộng không cần thiết. Từ góc nhìn của luận văn, đây là nhóm chỉ số rất quan trọng, bởi DWSA-VNF được thiết kế theo hướng không chỉ chấp nhận nhiều yêu cầu hơn mà

còn khai thác tài nguyên hợp lý hơn.

2.4.5. Yêu cầu về khả năng mở rộng và thích ứng động

Một đặc điểm nổi bật của môi trường NFV là tính động của hệ thống. Lưu lượng có thể thay đổi theo thời gian, dịch vụ mới có thể xuất hiện, dịch vụ cũ có thể kết thúc, một số VNF có thể cần scale-out hoặc scale-in, và một số node hạ tầng có thể rơi vào trạng thái quá tải hoặc lỗi. Do đó, một hệ thống NFV hiệu quả không chỉ cần đạt hiệu năng tốt ở trạng thái tĩnh, mà còn phải có khả năng thích ứng khi điều kiện vận hành thay đổi. Yêu cầu này thường được thể hiện thông qua các khái niệm như *khả năng mở rộng* (scalability), *khả năng thích ứng* (adaptability) và *tính đàn hồi* (elasticity) [1, 3].

Trong các nghiên cứu gần đây, workload theo thời gian được xem là một nhân tố quan trọng của tính động này, trong đó nhấn mạnh rằng placement nếu chỉ dựa trên một mức tải tĩnh sẽ khó khai thác tài nguyên hiệu quả trong các môi trường mà đỉnh tải giữa các dịch vụ không trùng nhau. Khi workload biến thiên theo thời gian, thông tin về profile tải có thể trở thành một tín hiệu quan trọng để quyết định colocate hay tách các VNF trên hạ tầng [11]. Đây cũng chính là một cơ sở quan trọng cho hướng tiếp cận của luận văn ở Chương 3.

2.4.6. Yêu cầu về khả năng chia sẻ, cô lập và quản lý trạng thái

Khi nhiều dịch vụ cùng chạy trên một hạ tầng NFV dùng chung, một yêu cầu kỹ thuật khác là làm sao cân bằng giữa *dùng chung tài nguyên* (resource sharing) và *cô lập dịch vụ* (service isolation). Nếu mọi VNF đều được tạo mới hoàn toàn, hệ thống có thể lãng phí tài nguyên vì overhead nền của mỗi instance. Ngược lại, nếu mọi VNF cùng loại đều được dùng chung một cách cứng nhắc, hệ thống có thể gặp rủi ro về trạng thái, chính sách hoặc mức độ an toàn. Vấn đề này đặc biệt rõ đối với các VNF stateful hoặc các VNF có xử lý phụ thuộc mạnh vào phiên và ngữ cảnh gói tin [1, 12].

Do đó, nhiều công trình đã bắt đầu xem xét các hướng *đa thuê* (multi-tenancy), *dùng chung instance* và *chia sẻ VNF* như một thành phần quan trọng trong tối ưu NFV. Tuy nhiên, sử dụng cơ chế sharing hiệu quả đòi hỏi phải xét tới loại VNF, profile tải, mức độ phụ thuộc trạng thái và mục tiêu QoS. Đây chính là lý do luận văn lựa chọn cách tiếp cận *chia sẻ có chọn lọc* (selective sharing) ở Chương 3 thay vì một chính sách share-all. Việc trình bày yêu cầu kỹ thuật này trong Chương 2 giúp làm rõ rằng selective sharing không phải là một ý tưởng ngẫu hứng, mà là phản ứng có cơ sở trước một yêu cầu thực sự của môi trường NFV.

Một hệ quả quan trọng của việc tồn tại đồng thời nhiều yêu cầu kỹ thuật như trên là không có một chỉ số đơn lẻ nào đủ để phản ánh đầy đủ chất lượng của một phương pháp cải tiến hiệu năng trong NFV. Một thuật toán có thể đạt *tỷ lệ chấp nhận yêu cầu* (acceptance ratio) cao nhưng lại làm tăng mạnh *độ trễ đầu cuối* (end-to-end delay). Một thuật toán khác có thể duy trì delay thấp nhưng sử dụng tài nguyên quá phân tán hoặc tiêu tốn nhiều overhead cho việc khởi tạo instance. Cũng có những trường hợp một phương pháp cho kết quả tốt về tài nguyên nhưng lại không thích ứng tốt khi workload thay đổi theo thời gian [7, 9, 11]. Chính vì vậy, khi đánh giá các giải pháp placement/chaining trong NFV, cần nhìn nhận hiệu năng như một bài toán đa mục tiêu và đa ràng buộc.

Nhận định này đặc biệt quan trọng đối với luận văn, bởi nó giải thích tại sao các chỉ số được dùng ở Chương 4 không chỉ bao gồm một đại lượng duy nhất. Thay vào đó, luận văn lựa chọn đồng thời các chỉ số về khả năng phục vụ, chất lượng dịch vụ, hiệu quả sử dụng tài nguyên và chi phí thực thi. Cách tiếp cận này bám sát bản chất kỹ thuật của môi trường NFV/SFC và giúp việc đánh giá phương pháp đề xuất ở các chương sau có cơ sở học thuật vững chắc hơn.

Từ các phân tích trên có thể thấy rằng môi trường NFV/SFC đặt ra một tập yêu cầu kỹ thuật khá phức tạp. Hệ thống không chỉ cần đủ tài nguyên, mà còn phải giữ delay trong ngưỡng, duy trì throughput và acceptance ratio ở mức tốt, sử dụng tài nguyên hiệu quả, thích ứng với workload động và xử lý hợp lý mối quan hệ giữa sharing với isolation. Chính sự đa chiều này làm cho bài toán placement trong NFV trở thành một bài toán khó, đồng thời cũng lý giải vì sao có nhiều nhóm phương pháp cải tiến hiệu năng khác nhau đã được nghiên cứu trong tài liệu. Trên nền đó, phần tiếp theo của chương sẽ khảo sát các nhóm phương pháp chính đang được sử dụng để giải quyết bài toán này.

2.5. Các nhóm phương pháp cải tiến hiệu năng NFV

Trên nền tảng kiến trúc và các yêu cầu kỹ thuật đã trình bày ở các mục trước, có thể thấy rằng cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization) không phải là một bài toán đơn lẻ, mà là một tập hợp các bài toán liên quan chặt chẽ với nhau. Trong các khảo sát gần đây, các hướng nghiên cứu về hiệu năng NFV thường được tổ chức thành một số nhóm lớn như *phân bổ tài nguyên và điều phối dịch vụ* (resource allocation and service orchestration), *đặt và khâu chuỗi chức năng mạng ảo* (VNF placement and chaining), *tối ưu hiệu năng thực*

thi (performance tuning), và *khả năng phục hồi cùng khôi phục lỗi* (resilience and fault recovery) [1, 3, 7, 9]. Việc khảo sát theo nhóm như vậy giúp làm rõ bức tranh nghiên cứu tổng thể, đồng thời cho phép xác định chính xác nhóm phương pháp nào có liên hệ trực tiếp nhất với đề tài của luận văn.

2.5.1. Nhóm phương pháp phân bổ tài nguyên và điều phối dịch vụ

Phân bổ tài nguyên (resource allocation) là một trong những hướng nghiên cứu sớm và quan trọng nhất của NFV. Khi nhiều VNF cùng chia sẻ một hạ tầng chung, hệ thống phải quyết định mỗi dịch vụ nên được cấp bao nhiêu *bộ xử lý trung tâm* (CPU), bộ nhớ, băng thông, và được phân phối trên những thành phần nào của hạ tầng. Vấn đề này không chỉ liên quan đến hiệu quả sử dụng tài nguyên, mà còn ảnh hưởng trực tiếp đến *chất lượng dịch vụ* (Quality of Service), độ trễ đầu cuối, thông lượng, và khả năng mở rộng của hệ thống [7, 9].

Các nghiên cứu trong nhóm này thường xem tài nguyên như một đối tượng tối ưu hóa. Một số công trình tập trung vào việc xây dựng mô hình tối ưu hóa nguyên tuyến tính hoặc biến thể của nó để tìm lời giải phân bổ tài nguyên tối ưu theo các ràng buộc đã biết trước [7, 24]. Một số hướng khác sử dụng heuristic hoặc metaheuristic để giảm chi phí tính toán khi số lượng node, số lượng yêu cầu hoặc độ dài chuỗi dịch vụ tăng lên [8, 9]. Trong các bối cảnh động hơn, một số tác giả còn kết hợp dự báo tải hoặc học máy để hỗ trợ quyết định phân bổ tài nguyên theo thời gian [9, 14].

Gắn liền với phân bổ tài nguyên là *điều phối dịch vụ* (service orchestration). Nếu phân bổ tài nguyên trả lời câu hỏi một dịch vụ được cấp bao nhiêu tài nguyên, thì orchestration trả lời câu hỏi dịch vụ đó được triển khai ở đâu, theo thứ tự nào, trên chuỗi xử lý nào và được thay đổi ra sao trong suốt vòng đời vận hành. Các công trình về orchestration thường nghiên cứu logic điều phối ở cấp độ toàn hệ thống, bao gồm placement, chaining, scheduling, monitoring, scaling và reconfiguration [3, 19, 22]. Điều này cho thấy resource allocation và orchestration thực chất là hai mặt của cùng một bài toán quản trị hiệu năng trong NFV.

2.5.2. Nhóm phương pháp đặt và xâu chuỗi VNF

Trong tất cả các nhóm phương pháp, *đặt VNF* (VNF placement) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining) là nhóm có liên hệ trực tiếp nhất đến đề tài của luận văn. Placement quyết định mỗi VNF được đặt trên node nào của hạ tầng, còn chaining quyết định các VNF được liên kết ra sao để tạo thành dịch vụ hoàn chỉnh. Trong môi trường NFV hiện đại, hai bài toán này hầu như không thể tách rời

nhau, vì vị trí của từng VNF sẽ ảnh hưởng trực tiếp đến đường đi của lưu lượng, đến độ trễ tích lũy và đến mức sử dụng băng thông trên toàn hệ thống [3, 5, 8].

Một số nghiên cứu tiếp cận placement như một bài toán tối ưu hóa tĩnh, trong đó topology, tài nguyên và tập yêu cầu được giả định đã biết trước. Mục tiêu của các mô hình này thường là giảm chi phí triển khai, giảm độ trễ hoặc cân bằng tải [21, 24]. Tuy nhiên, khi quy mô bài toán tăng lên, các lời giải tối ưu chính xác thường gặp khó khăn về chi phí tính toán. Đây là lý do khiến ngày càng nhiều công trình chuyển sang sử dụng heuristic, approximation algorithm hoặc các mô hình học tăng cường để tìm lời giải đủ tốt trong thời gian chấp nhận được [8, 9, 15].

Một hướng phát triển đáng chú ý của nhóm này là mở rộng placement từ trung tâm dữ liệu truyền thống sang các môi trường biên và liên đám mây. Khi dịch vụ được triển khai trên không gian hạ tầng phân tán hơn, bài toán placement không chỉ cần xét tài nguyên mà còn phải xét khoảng cách, độ trễ truyền dẫn và sự khác biệt về năng lực hạ tầng giữa các miền [13, 25]. Những xu hướng này cho thấy placement/chaining vẫn là một hướng nghiên cứu rất mở, đặc biệt trong các bối cảnh workload động, edge computing hoặc mạng hỗn hợp cloud–edge.

2.5.3. Nhóm phương pháp tối ưu hiệu năng thực thi

Bên cạnh placement và resource allocation, một nhánh nghiên cứu khác tập trung vào *tối ưu hiệu năng thực thi* (performance tuning) của chính các nền tảng NFV và các VNF đang chạy. Khác với các phương pháp quyết định “đặt cái gì ở đâu”, nhóm này quan tâm nhiều hơn đến việc “chạy như thế nào cho hiệu quả hơn” trên cùng một nền hạ tầng. Các chủ đề thường gặp bao gồm tối ưu packet I/O, giảm overhead của lớp ảo hóa, tối ưu lịch thực thi, batching, zero-copy, cache locality, NUMA-awareness và các cơ chế tăng tốc datapath [1, 18, 23].

Trong một số trường hợp, hiệu năng thấp không đến từ placement không hợp lý, mà đến từ cách VNF tương tác với hạ tầng bên dưới. Chẳng hạn, nếu packet I/O không được tối ưu, hoặc nếu tài nguyên CPU bị phân mảnh và tranh chấp cache nghiêm trọng, hệ thống vẫn có thể cho kết quả kém ngay cả khi placement về mặt logic là tương đối tốt. Chính vì vậy, nhiều khảo sát hiện nay đều nhấn mạnh rằng hiệu năng NFV là kết quả của cả hai lớp: lớp quyết định placement/chaining và lớp tối ưu execution/runtime [1, 18].

Từ góc nhìn của luận văn, nhóm phương pháp này không phải là trọng tâm trực tiếp, vì đề tài chủ yếu tập trung vào placement. Tuy nhiên, việc khảo sát nhóm này

vẫn cần thiết vì nó nhắc lại một nguyên tắc quan trọng: kết quả của một thuật toán placement luôn phải được diễn giải trong bối cảnh của nền tảng thực thi. Nói cách khác, placement chỉ là một phần của bài toán cải tiến hiệu năng tổng thể, dù trong phạm vi luận văn nó là phần được ưu tiên nghiên cứu sâu hơn.

2.5.4. Nhóm phương pháp tăng độ tin cậy, khả năng phục hồi và khôi phục lỗi

Một hướng nghiên cứu khác cũng được nhiều tài liệu nhấn mạnh là *khả năng phục hồi và khôi phục lỗi* (resilience and fault recovery). Trong môi trường NFV, khi các dịch vụ được xây dựng từ nhiều VNF chạy trên hạ tầng dùng chung, rủi ro hỏng hóc hoặc lỗi ở một node vật lý có thể ảnh hưởng đến nhiều dịch vụ đồng thời. Điều này làm xuất hiện nhu cầu phải có các cơ chế backup, failover, migration và redundancy phù hợp [3, 12].

Các nghiên cứu trong nhóm này thường tập trung vào việc bảo đảm tính sẵn sàng của dịch vụ, giảm thời gian gián đoạn và duy trì SLA khi xuất hiện sự cố. Một số công trình đề xuất các mô hình redundancy có dùng chung dự phòng giữa nhiều tenant, trong khi một số khác quan tâm đến placement có xét availability như một ràng buộc cùng với tài nguyên và delay [10, 12]. Mặc dù đây là một hướng rất quan trọng đối với vận hành thực tế, nó cũng làm tăng đáng kể độ phức tạp của bài toán.

Trong khuôn khổ luận văn này, các cơ chế backup, migration và failover chưa được đưa vào mô hình chính của phương pháp đề xuất. Tuy nhiên, việc khảo sát nhóm phương pháp này là cần thiết để giới hạn rõ phạm vi nghiên cứu: luận văn tập trung vào placement/chaining và tối ưu hiệu năng khai thác tài nguyên, thay vì mở rộng sang các bài toán bảo đảm độ tin cậy ở mức đầy đủ.

2.5.5. Nhận xét tổng hợp về các nhóm phương pháp

Từ các nhóm phương pháp trên có thể thấy rằng nghiên cứu về cải tiến hiệu năng NFV là một không gian rất rộng. Một số hướng tập trung vào tài nguyên và orchestration, một số hướng tập trung vào placement/chaining, một số hướng đi sâu vào runtime/platform optimization, trong khi một số hướng khác nhấn mạnh resilience và availability. Điều đó cho thấy không có một lời giải đơn lẻ nào có thể bao quát toàn bộ nhu cầu hiệu năng của môi trường NFV [1, 3].

Đối với luận văn này, nhóm phương pháp có ý nghĩa trực tiếp nhất là placement/chaining, nhưng placement ở đây không được nhìn một cách tách rời khỏi resource allocation, delay, workload động hay cơ chế chia sẻ VNF. Chính vì vậy, phần tiếp theo của chương sẽ không dừng lại ở mức taxonomy tổng quát, mà sẽ đi sâu vào các

hướng nghiên cứu gần nhất với đề tài, từ đó chỉ ra khoảng trống mà phương pháp DWSA-VNF sẽ hướng đến trong Chương 3.

Nếu xét trong bốn nhóm phương pháp đã trình bày, có thể thấy *đặt và xâu chuỗi VNF* (VNF placement and chaining) là nhóm gần nhất với mục tiêu nghiên cứu của luận văn. Lý do là trong nhóm này, các yêu cầu về tài nguyên, độ trễ, throughput, topology, cấu trúc dịch vụ và logic điều phối gặp nhau trực tiếp nhất. Một quyết định placement không chỉ là quyết định chọn node, mà đồng thời là quyết định về việc sử dụng tài nguyên tính toán, sử dụng tài nguyên liên kết, hình thành đường đi của lưu lượng và giới hạn không gian placement của các VNF còn lại trong cùng chuỗi dịch vụ [3, 8, 21].

Chính vì vậy, mặc dù luận văn vẫn khảo sát các nhóm phương pháp khác để bảo đảm cái nhìn tổng thể, trọng tâm nghiên cứu cuối cùng vẫn được đặt vào placement/chaining. Đây cũng là lý do vì sao phần tiếp theo của chương sẽ không dừng ở mức taxonomy rộng, mà sẽ đi sâu vào các hướng placement có xét delay, workload theo thời gian, cơ chế dùng chung instance và các heuristic gần với bối cảnh ứng dụng của luận văn.

2.6. Phân tích các hướng liên quan và khoảng trống nghiên cứu

Sau khi đã khảo sát các nhóm phương pháp cải tiến hiệu năng NFV ở mức khái quát, phần này tập trung thu hẹp phạm vi vào những hướng nghiên cứu có liên hệ trực tiếp nhất tới đề tài của luận văn. Cụ thể, đó là các hướng placement/chaining có xét *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload), *đa thuê* (multi-tenancy), cơ chế dùng chung VNF instance và các heuristic thay cho lời giải tối ưu hóa chính xác. Đây cũng chính là lớp tài liệu quan trọng nhất để hình thành cơ sở khoa học cho phương pháp DWSA-VNF được đề xuất ở Chương 3.

2.6.1. Các hướng placement có xét độ trễ và yêu cầu SLA

Một trong những hướng được nghiên cứu nhiều nhất trong NFV/SFC là placement có xét đến delay và *cam kết mức dịch vụ* (Service Level Agreement). Lý do là trong nhiều môi trường dịch vụ, placement chỉ có ý nghĩa nếu chuỗi VNF sau khi được ánh xạ vẫn thỏa mãn được ngưỡng delay cho phép. Do đó, nhiều công trình không chỉ tối ưu tài nguyên, mà còn gắn placement với các ràng buộc về độ trễ đầu cuối, độ dài đường đi hoặc khoảng cách giữa các VNF [16, 21, 24].

Ưu điểm của nhóm hướng này là tạo ra các placement có giá trị vận hành thực tế hơn, tránh tình trạng chấp nhận được nhiều yêu cầu nhưng phần lớn trong số đó

không đạt điều kiện QoS. Tuy nhiên, hạn chế thường gặp là khi vừa phải thỏa tài nguyên vừa phải thỏa delay, không gian lời giải bị thu hẹp rất nhanh và chi phí tối ưu hóa tăng lên đáng kể. Đây là một lý do khiến nhiều tác giả phải chuyển từ lời giải chính xác sang heuristic hoặc approximation trong các bài toán quy mô lớn [8, 9].

2.6.2. Các hướng placement có xét workload động theo thời gian

Một phát triển mới đáng chú ý trong các nghiên cứu gần đây là việc đưa *tải biến thiên theo thời gian* (time-varying workloads) vào mô hình placement. Khác với các mô hình tĩnh, các hướng nghiên cứu này cho rằng nhu cầu xử lý của dịch vụ không nên được đại diện bằng một giá trị cố định duy nhất. Thay vào đó, profile tải của dịch vụ có thể thay đổi theo chu kỳ hoặc theo hành vi người dùng, và chính sự biến động đó cần được tận dụng như một tín hiệu để quyết định placement hiệu quả hơn [11, 14].

Nghiên cứu của Yi Yue và các cộng sự (2024) là một tài liệu đặc biệt gần với đề tài của luận văn. Công trình này chỉ ra rằng nếu xét đến workload theo thời gian, hệ thống có thể placement hiệu quả hơn bằng cách kết hợp những yêu cầu có đỉnh tải lệch pha trên cùng một node, từ đó tăng *tỷ lệ chấp nhận yêu cầu* (acceptance ratio) và *thông lượng mạng* (network throughput) [11]. Ý tưởng này rất quan trọng vì nó cho thấy placement hiệu quả không chỉ là placement trên tài nguyên tức thời, mà còn là placement có nhận thức về xu hướng tải. Tuy nhiên, nếu phát triển quá sâu, các mô hình kiểu này cũng dễ trở nên phức tạp và khó hiện thực trong môi trường nghiên cứu ứng dụng. Đây là điểm mà luận văn sẽ đơn giản hóa trong Chương 3: chỉ sử dụng workload theo thời gian như một tín hiệu hỗ trợ ra quyết định, thay vì xây dựng một mô hình dự báo động đầy đủ.

2.6.3. Các hướng nghiên cứu về đa thuê, sharing và phụ tải tài nguyên nền

Một hướng khác có liên hệ trực tiếp đến đề tài là nghiên cứu về *đa thuê* (multi-tenancy), dùng chung instance và *phụ tải tài nguyên nền* (Fundamental Resource Overhead). Khi mỗi yêu cầu dịch vụ đều khởi tạo instance riêng cho từng VNF, hệ thống có thể nhanh chóng tiêu tốn nhiều CPU, bộ nhớ và overhead ảo hóa, ngay cả khi tải dịch vụ thực sự chưa quá cao. Do đó, nhiều nghiên cứu đã tìm cách tận dụng shared redundancy hoặc dùng chung VNF instance để giảm chi phí này [11, 12].

Nghiên cứu của Yi Yue và các cộng sự (2024) tiếp cận vấn đề này theo hướng kết hợp multi-tenancy với time-varying workloads và FRO, cho thấy có thể tăng hiệu quả placement nếu hệ thống cho phép những yêu cầu phù hợp dùng chung instance trên cùng một node [11]. Tuy nhiên, một hạn chế của nhiều hướng sharing hiện nay là

thường giả định rằng mọi VNF cùng loại đều có thể reuse theo cùng một cách. Trên thực tế, điều này không phải lúc nào cũng hợp lý, đặc biệt với các VNF stateful hoặc nhạy với policy. Chính khoảng trống này mở ra cơ hội cho hướng *chia sẻ có chọn lọc* (selective sharing), tức không dùng chung một cách cứng nhắc cho mọi loại VNF, mà phân biệt theo đặc trưng kỹ thuật của từng nhóm chức năng.

2.6.4. Các hướng heuristic thay cho tối ưu hóa chính xác

Một đặc điểm phổ biến của bài toán placement/chaining trong NFV là độ phức tạp tính toán rất cao. Khi phải xét đồng thời topology, tài nguyên node, tài nguyên liên kết, delay, workload động và cấu trúc chuỗi dịch vụ, bài toán thường trở nên khó giải bằng các phương pháp tối ưu hóa chính xác ở quy mô lớn. Vì vậy, nhiều công trình gần đây lựa chọn hướng heuristic, approximation algorithm hoặc học tăng cường để đạt được chất lượng lời giải đủ tốt trong thời gian chấp nhận được [8, 9, 13, 15].

Ưu điểm lớn nhất của heuristic là tính khả thi trong hiện thực. Thay vì cố gắng tìm nghiệm tối ưu toàn cục, heuristic tập trung vào việc xây dựng một logic quyết định đủ hợp lý và đủ nhanh để có thể áp dụng trong các kịch bản có quy mô vừa và lớn. Trong bối cảnh luận văn thạc sĩ theo hướng ứng dụng, đây là một lựa chọn rất phù hợp. Nó cho phép người nghiên cứu cân bằng giữa tính học thuật và tính triển khai, đồng thời giúp môi trường mô phỏng trong lập trình Python vẫn giữ được sự gọn nhẹ cần thiết.

2.6.5. Khoảng trống nghiên cứu và định hướng của luận văn

Từ các hướng nghiên cứu vừa phân tích, có thể rút ra một số nhận xét quan trọng. Thứ nhất, placement/chaining tiếp tục là một trong những bài toán trung tâm của NFV và là hướng có ảnh hưởng trực tiếp nhất đến hiệu năng hệ thống. Thứ hai, nhiều công trình đã chỉ ra vai trò của delay, workload theo thời gian, multi-tenancy và resource efficiency, nhưng không phải lúc nào cũng kết hợp được các yếu tố này trong một mô hình vừa đủ đơn giản để hiện thực hóa. Thứ ba, các hướng sharing hiện có thường thiên về share-all hoặc thiên về mô hình tối ưu hóa khá nặng, trong khi còn thiếu các heuristic thực dụng theo hướng *chia sẻ có chọn lọc* (selective sharing) phù hợp với đặc tính khác nhau giữa các loại VNF [8, 11, 12].

Có thể thấy rằng khoảng trống nghiên cứu mà luận văn hướng tới không nằm ở việc phủ nhận các hướng nghiên cứu hiện có, mà nằm ở việc lựa chọn một điểm cân bằng phù hợp hơn giữa tính chặt chẽ học thuật và khả năng triển khai thực tế trong môi trường nghiên cứu ứng dụng. Các mô hình tối ưu hóa chính xác thường rất mạnh

về mặt hình thức, nhưng dễ trở nên khó hiện thực hoặc khó mở rộng khi topology, số lượng yêu cầu và số lượng ràng buộc tăng lên. Ngược lại, các baseline đơn giản tuy dễ triển khai nhưng thường bỏ qua những yếu tố quan trọng như delay, workload động hoặc sự khác biệt giữa các loại VNF trong cơ chế dùng chung. Chính khoảng giữa của hai cực này là nơi một heuristic có thiết kế tốt có thể phát huy giá trị [8, 9, 11].

Từ góc nhìn đó, luận văn không đặt mục tiêu xây dựng một lời giải tối ưu hóa hình thức mới, mà hướng tới một phương pháp vừa đủ đơn giản để hiện thực trong lập trình Python, vừa đủ mạnh để thể hiện được lợi ích của việc kết hợp nhiều tiêu chí kỹ thuật vào cùng một quy trình placement/chaining. Đây là điểm làm cho đề tài vừa phù hợp với phạm vi của một luận văn thạc sĩ, vừa vẫn giữ được giá trị học thuật và giá trị thực nghiệm.

Chính từ những nhận xét trên, luận văn lựa chọn hướng nghiên cứu là xây dựng một heuristic placement có xét đồng thời *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và selective sharing. So với các hướng tối ưu hóa đầy đủ, phương pháp này được định hướng theo tinh thần kỹ thuật ứng dụng: giảm bớt độ phức tạp toán học, tăng khả năng hiện thực trong môi trường lập trình Python, nhưng vẫn giữ được logic khoa học đủ rõ để có thể đánh giá định lượng. Đây chính là cơ sở trực tiếp để luận văn chuyển sang Chương 3, nơi phương pháp DWSA-VNF sẽ được xây dựng và mô tả chi tiết.

2.7. Kết luận chương

Chương này đã khảo sát cơ sở lý thuyết và các hướng nghiên cứu chính liên quan đến cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization). Trước hết, luận văn đã trình bày tổng quan về mối quan hệ giữa NFV, *mạng điều khiển bằng phần mềm* (Software-Defined Networking) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), qua đó xác lập nền công nghệ của bài toán placement/chaining. Tiếp theo, chương đã phân tích kiến trúc và các thành phần chính của hệ thống NFV, bao gồm *hạ tầng NFV* (NFVI), *quản lý và điều phối NFV* (MANO), *bộ điều phối NFV* (NFVO), *bộ quản lý hạ tầng ảo hóa* (VIM) và *bộ quản lý chức năng mạng ảo* (VNFM), đồng thời làm rõ các lựa chọn thiết kế nền tảng có ảnh hưởng trực tiếp đến hiệu năng.

Trên nền tảng đó, chương đã thảo luận các yêu cầu kỹ thuật và các chỉ tiêu hiệu năng trong môi trường NFV/SFC, bao gồm tài nguyên tính toán, tài nguyên mạng, delay, throughput, resource efficiency, khả năng thích ứng động và mối quan hệ giữa

sharing với isolation. Sau đó, các nhóm phương pháp cải tiến hiệu năng NFV đã được khảo sát theo bốn hướng lớn là resource allocation and service orchestration, VNF placement and chaining, performance tuning, và resilience/fault recovery. Cuối cùng, chương tập trung phân tích sâu hơn các hướng nghiên cứu liên quan trực tiếp đến đề tài, đặc biệt là các hướng placement có xét delay, workload theo thời gian, multi-tenancy và heuristic thay cho tối ưu hóa chính xác.

Từ quá trình khảo sát này, luận văn xác định được một khoảng trống nghiên cứu phù hợp: cần có một phương pháp đặt chỗ (placement) vừa đủ đơn giản để hiện thực trong môi trường mô phỏng, nhưng vẫn xét đồng thời các yếu tố quan trọng như *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing). Trên cơ sở đó, Chương 3 sẽ chuyển từ phần khảo sát tài liệu sang phần đề xuất phương pháp, trong đó luận văn xây dựng và trình bày chi tiết phương pháp DWSA-VNF như đóng góp chính của đề tài.

CHƯƠNG 3. ĐỀ XUẤT PHƯƠNG PHÁP CẢI TIẾN HIỆU NĂNG CHO MÔI TRƯỜNG NFV

3.1. Giới thiệu chương

Trong Chương 1 và Chương 2, luận văn đã trình bày nền tảng của công nghệ NFV và khảo sát các hướng nghiên cứu liên quan đến cải tiến hiệu năng trong môi trường *ảo hóa chức năng mạng* (Network Function Virtualization). Kết quả khảo sát cho thấy bài toán đặt và xâu chuỗi *chức năng mạng ảo* (Virtual Network Function) giữ vai trò trung tâm, bởi quyết định placement ảnh hưởng trực tiếp đến khả năng khai thác tài nguyên của hạ tầng, *độ trễ đầu cuối* (end-to-end delay), *thông lượng mạng* (network throughput) và *tỷ lệ chấp nhận yêu cầu* (acceptance ratio) của hệ thống [3, 7, 11].

Tuy nhiên, phần lớn các hướng tiếp cận hiện có hoặc khá phức tạp về mặt mô hình tối ưu hóa, hoặc khó thực nghiệm đầy đủ trong một môi trường mô phỏng trong phạm vi một luận văn. Trong khi đó, mục tiêu của đề tài luận văn không phải là xây dựng một lời giải tối ưu toàn cục có tính lý thuyết cao, mà là đề xuất một phương pháp có cơ sở kỹ thuật rõ ràng, có thể hiện thực được, có thể mô phỏng được, và có khả năng tạo ra kết quả định lượng để đánh giá. Vì vậy, chương này lựa chọn một hướng tiếp cận thiên về kỹ thuật, trong đó phương pháp đề xuất được xây dựng như một heuristic placement/chaining phù hợp với môi trường NFV có xét đến tài nguyên, độ trễ, đặc tính tải theo thời gian và cơ chế chia sẻ VNF instance.

Trên cơ sở đó, luận văn đề xuất phương pháp DWSA-VNF, viết tắt của *đặt VNF có xét độ trễ, tải biến thiên theo thời gian và chia sẻ chọn lọc* (Delay and Workload-aware Selective Aggregation for VNF Placement). Ý tưởng cốt lõi của phương pháp này là kết hợp ba yếu tố thường có ảnh hưởng lớn đến hiệu năng NFV, bao gồm ràng buộc delay, workload theo thời gian và cơ chế *chia sẻ có chọn lọc* (selective sharing), thay vì áp dụng các chính sách placement hoặc reuse mang tính cứng nhắc.

Nội dung còn lại của chương được tổ chức như sau. Mục 3.2 trình bày cơ sở đề xuất phương pháp và bài toán đặt ra. Mục 3.3 mô tả mô hình hệ thống và các giả thiết mô phỏng. Mục 3.4 trình bày thiết kế của phương pháp DWSA-VNF. Mục 3.5 trình bày môi trường cài đặt và tổ chức mô phỏng, làm cơ sở trực tiếp cho phần đánh giá

thực nghiệm ở Chương 4. Cuối cùng, Mục 3.6 đưa ra kết luận chương.

3.2. Cơ sở đề xuất phương pháp và bài toán đặt ra

Cơ sở của phương pháp đề xuất trong luận văn xuất phát từ thực tế rằng hiệu năng của hệ thống NFV không chỉ phụ thuộc vào năng lực của hạ tầng vật lý, mà còn phụ thuộc rất mạnh vào cách các VNF được đặt và xâu chuỗi trên hạ tầng đó. Trong môi trường *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), một yêu cầu dịch vụ chỉ được xem là phục vụ thành công nếu toàn bộ chuỗi VNF của nó được ánh xạ đầy đủ, đúng thứ tự và vẫn thỏa mãn các ràng buộc về tài nguyên và *chất lượng dịch vụ* (Quality of Service). Điều này khiến placement trở thành một bài toán khó, phải xét đồng thời tài nguyên node, tài nguyên liên kết, cấu trúc chuỗi dịch vụ và yêu cầu *cam kết mức dịch vụ* (Service Level Agreement) [3, 7].

Các nghiên cứu gần đây cho thấy những hướng như *đặt động VNF* (dynamic VNF placement), *điều hướng lưu lượng* (traffic steering), *nhúng VNF/SFC* (VNF/SFC embedding) và bảo đảm SLA vẫn là các hướng mở quan trọng của lĩnh vực [3]. Trong khi đó, nhiều phương pháp cơ bản như *đặt ngẫu nhiên* (Random Placement) hoặc *tham lam ưu tiên nút đầu tiên phù hợp* (Greedy First-Fit) tuy dễ cài đặt nhưng thường bộc lộ hạn chế khi số lượng yêu cầu tăng lên hoặc khi điều kiện độ trễ trở nên chặt chẽ hơn. Những phương pháp này thường chỉ tối ưu cục bộ theo một tiêu chí hẹp, chẳng hạn tài nguyên còn lại của node hoặc vị trí gần nhất về mặt topology, nên khó tận dụng tốt hạ tầng trong các kịch bản phức tạp hơn.

Bên cạnh các ràng buộc tài nguyên và cấu trúc chuỗi dịch vụ, một yếu tố quan trọng khác là đặc tính động của tải hệ thống. Trong nhiều mô hình placement đơn giản, mỗi yêu cầu thường được biểu diễn bởi một mức tải tĩnh. Tuy nhiên, trong môi trường thực tế, lưu lượng dịch vụ thường thay đổi theo thời gian. Có nghiên cứu đã cho thấy rằng nếu xét đến *tải biến thiên theo thời gian* (time-varying workloads), placement có thể đạt hiệu quả tốt hơn khi biết khai thác sự khác biệt trong mẫu tải của các yêu cầu dịch vụ [11]. Nhận định này đặc biệt phù hợp với mục tiêu của luận văn, bởi nó mở ra một hướng cải tiến vừa có cơ sở học thuật, vừa đủ đơn giản để hiện thực hóa.

Một cơ sở quan trọng khác dẫn đến việc đề xuất phương pháp mới là cơ chế dùng chung VNF instance. Về nguyên tắc, dùng chung có thể giúp giảm *phụ tải tài nguyên nền* (Fundamental Resource Overhead) do tránh phải tạo quá nhiều instance mới. Tuy nhiên, việc chia sẻ không nên áp dụng một cách đồng loạt cho mọi loại VNF. Các chức năng như NAT hoặc cân bằng tải thường phù hợp hơn với reuse, trong

khi các chức năng như *tường lửa* (firewall), *hệ thống phát hiện xâm nhập* (intrusion detection system) hay *kiểm tra gói tin sâu* (deep packet inspection) có thể nhạy hơn với trạng thái và chính sách riêng [1, 11]. Vì vậy, việc đưa *chia sẻ có chọn lọc* (selective sharing) vào phương pháp đề xuất là hợp lý hơn so với một chính sách *chia sẻ toàn phần* (Greedy Share-All) đơn giản.

Từ các phân tích trên, bài toán mà luận văn đặt ra có thể được phát biểu ngắn gọn như sau: với một hạ tầng NFV hữu hạn tài nguyên, một tập các yêu cầu dịch vụ xuất hiện dưới dạng chuỗi VNF, và các ràng buộc về tài nguyên, độ trễ cùng workload theo thời gian, cần xây dựng một phương pháp placement có khả năng tăng *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), cải thiện *thông lượng mạng* (network throughput), sử dụng tài nguyên hiệu quả hơn và vẫn duy trì chất lượng dịch vụ ở mức chấp nhận được. Trên cơ sở đó, luận văn đề xuất DWSA-VNF như một heuristic placement/chaining có xét đồng thời delay, workload động và selective sharing, qua đó tạo ra sự khác biệt rõ ràng so với các baseline đơn giản như Random Placement, Greedy First-Fit và Greedy Share-All.

3.3. Mô hình hệ thống và giả thiết mô phỏng

Trong luận văn này, hệ thống được mô hình hóa theo hướng đủ chặt chẽ để phục vụ thiết kế thuật toán và mô phỏng, nhưng không đi quá sâu vào hình thức toán học. Về tổng thể, môi trường nghiên cứu bao gồm một hạ tầng vật lý hữu hạn tài nguyên, một tập các yêu cầu dịch vụ được biểu diễn dưới dạng chuỗi VNF, các profile tải theo thời gian, cùng với các ràng buộc về tài nguyên và độ trễ.

Hạ tầng vật lý được biểu diễn dưới dạng một đồ thị:

$$G = (N, L) \quad (3.3.1)$$

trong đó N là tập node vật lý có khả năng thực thi VNF, còn L là tập liên kết vật lý giữa các node. Mỗi node có giới hạn về *bộ xử lý trung tâm* (CPU) và bộ nhớ, còn mỗi liên kết có giới hạn về băng thông và độ trễ truyền dẫn. Bên cạnh tài nguyên khả dụng, mỗi node còn có trạng thái hoạt động, phản ánh việc node đó đã được sử dụng để chạy ít nhất một VNF hay chưa. Thuộc tính này có vai trò quan trọng trong phương pháp DWSA-VNF, bởi thuật toán ưu tiên sử dụng các node đang hoạt động trước khi kích hoạt thêm node mới.

Mỗi yêu cầu dịch vụ trong mô hình được xem như một *yêu cầu chuỗi chức năng dịch vụ* (Service Function Chain Request), bao gồm một chuỗi VNF phải đi qua

theo đúng thứ tự, cùng với nhu cầu tài nguyên và ngưỡng độ trễ đầu cuối tối đa. Như vậy, một yêu cầu chỉ được coi là placement thành công nếu toàn bộ chuỗi được ánh xạ trọn vẹn lên hạ tầng vật lý và tổng độ trễ vẫn không vượt quá giới hạn đã đặt ra. Cách mô hình hóa này phản ánh đúng bản chất của bài toán placement/chaining trong NFV, nơi quyết định đối với một VNF luôn ảnh hưởng đến phần còn lại của chuỗi dịch vụ [3, 7].

Một điểm mở rộng đáng chú ý của mô hình là việc đưa *tải biến thiên theo thời gian* (time-varying workload) vào như một tín hiệu hỗ trợ placement. Trong phạm vi luận văn, profile tải theo time-slots được sử dụng như thông tin hỗ trợ ra quyết định placement, thay vì làm đầu vào cho một mô hình dự báo lưu lượng độc lập. Workload động không được dùng để xây dựng một mô hình dự báo phức tạp, mà được dùng để giúp thuật toán tránh đặt chồng những yêu cầu có xu hướng đạt đỉnh tải cùng lúc lên cùng một node. Cách tiếp cận này bám sát tinh thần của các nghiên cứu gần đây nhưng vẫn giữ được sự đơn giản cần thiết cho mô phỏng [11].

Bên cạnh đó, mô hình còn đưa vào cơ chế *chia sẻ có chọn lọc* (selective sharing). Cụ thể, một số loại VNF được xem là phù hợp hơn với việc dùng chung instance, chẳng hạn NAT, *cân bằng tải* (load balancer) và *tối ưu WAN* (WAN optimizer). Ngược lại, các VNF như *tường lửa* (firewall), *hệ thống phát hiện xâm nhập* (intrusion detection system) và *kiểm tra gói tin sâu* (deep packet inspection) được xếp vào nhóm không chia sẻ trong phạm vi mô hình của luận văn. Cách phân loại này không nhằm mô tả một chuẩn chung cho mọi hệ thống NFV, mà nhằm tạo ra một quy tắc kỹ thuật nhất quán để phục vụ thiết kế thuật toán [1, 11].

Để thu hẹp phạm vi nghiên cứu và bảo đảm khả năng mô phỏng, luận văn sử dụng một số giả thiết. Trước hết, topology vật lý được giả định là đã biết tại thời điểm placement. Tiếp theo, mỗi yêu cầu dịch vụ có chuỗi VNF cố định trong suốt quá trình mô phỏng. Luận văn chưa xét các cơ chế *dự phòng* (backup), *chuyển đổi dự phòng* (failover) hoặc *di chuyển VNF* (VNF migration), vì việc đưa các yếu tố này vào sẽ làm bài toán trở nên phức tạp hơn đáng kể. Ngoài ra, việc kết nối giữa các VNF trong chuỗi được giả định có thể thực hiện thông qua một tuyến khả dụng phù hợp với ràng buộc độ trễ, thay vì phải giải một bài toán *kỹ thuật lưu lượng* (traffic engineering) độc lập ở mỗi bước placement.

Để chuẩn hóa cách hiểu các thành phần của mô hình, Bảng 3.3.1 tóm tắt một số ký hiệu và tham số sử dụng trong chương.

Bảng 3.3.1. Ký hiệu và tham số chính của mô hình hệ thống

Ký hiệu	Ý nghĩa
$G = (N, L)$	Đồ thị hạ tầng vật lý
N	Tập node vật lý
L	Tập liên kết vật lý
R	Tập yêu cầu dịch vụ
S_r	Chuỗi VNF của yêu cầu r
T	Số lượng time-slots mô phỏng
$D_r^{\max}$	Độ trễ đầu cuối tối đa của yêu cầu r
C_u^{cpu}	Dung lượng CPU của node u
C_u^{mem}	Dung lượng bộ nhớ của node u
C_{uv}^{bw}	Băng thông của liên kết (u, v)

Bảng 3.3.2 trình bày quy tắc phân loại VNF theo khả năng chia sẻ trong phạm vi luận văn.

Bảng 3.3.2. Phân loại VNF theo khả năng chia sẻ trong mô hình

Loại VNF	Chia sẻ	Ghi chú
NAT	Có	Phù hợp với cơ chế dùng chung instance
Load Balancer	Có	Có thể reuse trong nhiều yêu cầu dịch vụ
WAN Optimizer	Có	Có thể tận dụng chia sẻ trong môi trường nhiều tải
Firewall	Không	Nhạy với trạng thái và chính sách riêng
IDS	Không	Phụ thuộc trạng thái xử lý và cấu hình giám sát
DPI	Không	Xử lý sâu, khó chia sẻ an toàn

Như vậy, mô hình hệ thống của luận văn đã xác định các thành phần cần thiết cho bài toán placement trong môi trường NFV, bao gồm hạ tầng vật lý, chuỗi dịch vụ, workload động và selective sharing. Trên nền tảng đó, phần tiếp theo của chương sẽ trình bày thiết kế của phương pháp DWSA-VNF, tức cách các thành phần của mô hình được chuyển hóa thành các nguyên tắc và quy trình ra quyết định cụ thể.

3.4. Thiết kế phương pháp DWSA-VNF

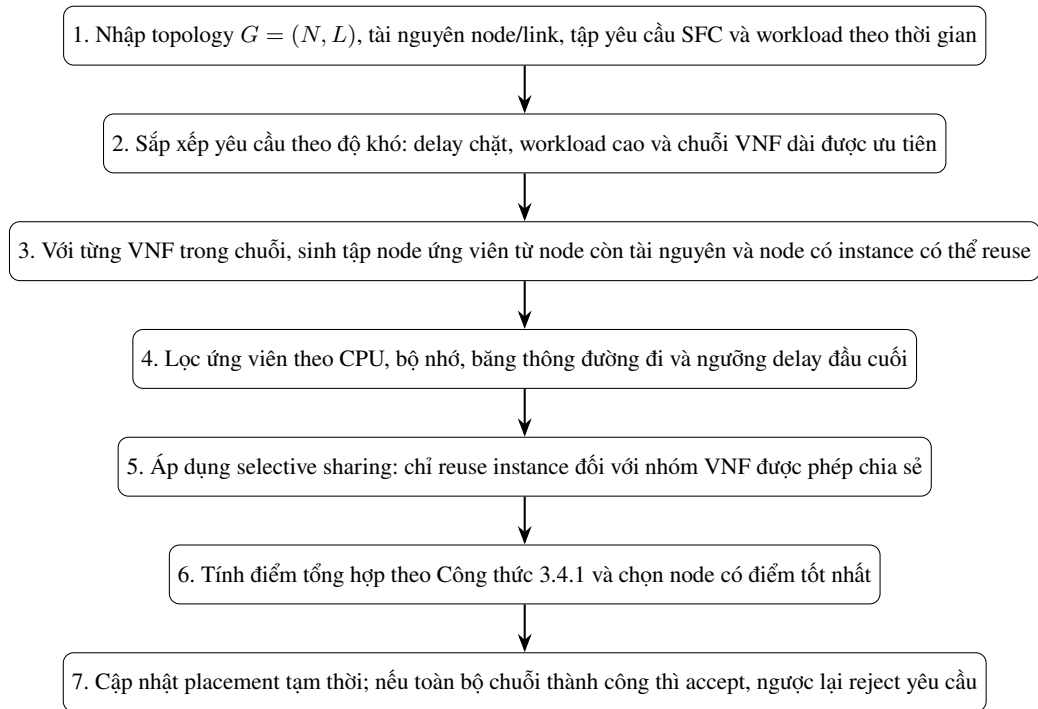
Phương pháp DWSA-VNF được xây dựng như một heuristic placement/chaining có định hướng kỹ thuật, trong đó quyết định placement không dựa trên một tiêu chí đơn lẻ mà dựa trên sự kết hợp của nhiều yếu tố như tài nguyên, độ trễ, trạng thái

hoạt động của node và khả năng chia sẻ VNF instance. Mục tiêu của phương pháp không phải là tìm nghiệm tối ưu toàn cục trong mọi trường hợp, mà là tạo ra một quy trình đủ tốt, có thể hiện thực hóa, có thể đánh giá bằng mô phỏng và phù hợp với bối cảnh của luận văn thạc sĩ [7, 11].

Đầu vào của DWSA-VNF bao gồm topology vật lý, trạng thái tài nguyên node và liên kết, tập yêu cầu dịch vụ, profile workload theo thời gian và ngưỡng độ trễ của từng yêu cầu. Đầu ra của thuật toán là quyết định placement cho từng VNF trong chuỗi, trạng thái chấp nhận hoặc từ chối của từng yêu cầu, tập instance được dùng chung hoặc tạo mới, cùng với trạng thái cập nhật của tài nguyên hệ thống sau quá trình ánh xạ. Nhờ cách xác định đầu vào và đầu ra như vậy, DWSA-VNF có thể được xem là một cơ chế ra quyết định tuần tự, trong đó kết quả của mỗi lần placement sẽ ảnh hưởng trực tiếp đến không gian quyết định cho các yêu cầu tiếp theo.

Tư tưởng thiết kế của DWSA-VNF dựa trên bốn định hướng chính. Thứ nhất, các yêu cầu dịch vụ không nên được xử lý hoàn toàn ngẫu nhiên, mà cần có một mức độ ưu tiên nhất định để giảm khả năng những yêu cầu khó bị loại bỏ quá sớm. Thứ hai, thuật toán cần sàng lọc trước các node không khả thi về tài nguyên hoặc độ trễ thay vì chấm điểm tất cả các phương án. Thứ ba, cơ chế dùng chung VNF chỉ nên áp dụng có chọn lọc để tránh sự đơn giản hóa quá mức của mô hình. Thứ tư, quyết định cuối cùng cần dựa trên một hàm điểm tổng hợp để cân bằng giữa các tiêu chí placement thay vì chỉ dựa vào quy tắc cứng như first-fit.

Để làm rõ hơn logic xử lý của phương pháp đề xuất, Hình 3.4.1 mô tả pipeline tổng thể của DWSA-VNF. Điểm cần nhấn mạnh là DWSA-VNF không chỉ bổ sung một hàm điểm vào thuật toán tham lam, mà tổ chức lại toàn bộ tiến trình placement theo nhiều tầng xử lý: ưu tiên yêu cầu, lọc khả thi, kiểm tra ràng buộc, xét khả năng chia sẻ và cuối cùng mới chấm điểm lựa chọn node. Pipeline này cũng là cơ sở để hiện thực phương pháp trong script mô phỏng và để giải thích các kết quả thực nghiệm ở Chương 4.



Hình 3.4.1. Pipeline tổng thể của phương pháp DWSA-VNF

Trước hết, DWSA-VNF thực hiện sắp xếp các yêu cầu dịch vụ theo mức độ ưu tiên. Trong phạm vi luận văn, mức độ ưu tiên được hiểu theo hướng thực dụng, chủ yếu dựa trên mức độ chặt của ràng buộc delay, độ dài chuỗi VNF và cường độ tải dự kiến. Ý tưởng của bước này là nếu các yêu cầu khó bị xử lý quá muộn, chúng rất dễ mất cơ hội placement do các vùng tài nguyên thuận lợi nhất đã bị các yêu cầu đơn giản hơn sử dụng trước. Bằng cách đưa một trật tự xử lý có chủ đích vào thuật toán, DWSA-VNF làm giảm sự phụ thuộc của kết quả vào thứ tự sinh dữ liệu đầu vào.

Sau bước sắp xếp, với từng yêu cầu dịch vụ được chọn, thuật toán xét lần lượt từng VNF trong chuỗi. Đối với mỗi VNF, DWSA-VNF sinh ra một tập node ứng viên. Tập này bao gồm những node còn khả năng chứa VNF mới, hoặc những node đang có sẵn instance cùng loại trong trường hợp selective sharing cho phép. Sau đó, thuật toán sàng lọc theo hai điều kiện nền tảng. Trước hết, các node không còn đủ CPU, bộ nhớ hoặc làm liên kết liên quan vượt quá giới hạn bảng thông sẽ bị loại bỏ. Tiếp theo, thuật toán kiểm tra xem việc đặt VNF tại node đó có làm tổng độ trễ của chuỗi dịch vụ vượt ngưỡng hay không. Hai bước sàng lọc này giúp cơ chế chấm điểm chỉ được áp dụng trên những phương án thực sự có ý nghĩa về mặt kỹ thuật.

Sau khi đã có tập node ứng viên hợp lệ, thuật toán xét cơ chế selective sharing. Nếu VNF đang được xử lý thuộc nhóm cho phép chia sẻ, DWSA-VNF ưu tiên những node đang có sẵn instance cùng loại và còn đủ khả năng phục vụ thêm. Nếu VNF

thuộc nhóm không chia sẻ, các phương án reuse bị loại bỏ và chỉ các phương án tạo mới instance được giữ lại. Thiết kế này giúp DWSA-VNF tận dụng được lợi ích của *đa thuê* (multi-tenancy) ở những nơi phù hợp, nhưng vẫn duy trì một mô hình placement có tính hợp lý kỹ thuật [1, 11].

Để lựa chọn giữa các node ứng viên còn lại, DWSA-VNF sử dụng một hàm điểm tổng hợp. Hàm điểm này không được trình bày như một mô hình tối ưu hóa đa mục tiêu đầy đủ, mà chỉ đóng vai trò kết hợp các tiêu chí chính thành một cơ chế so sánh thống nhất. Một dạng tổng quát của hàm điểm được sử dụng trong luận văn như sau:

$$Score(u) = \alpha S_{delay}(u) + \beta S_{resource}(u) + \gamma S_{reuse}(u) + \delta S_{share}(u) - \eta P_{new}(u) \quad (3.4.1)$$

trong đó $S_{delay}(u)$ phản ánh mức độ thuận lợi về độ trễ, $S_{resource}(u)$ phản ánh lượng tài nguyên còn dư sau placement, $S_{reuse}(u)$ phản ánh lợi ích của việc dùng node đang hoạt động, $S_{share}(u)$ phản ánh lợi ích của selective sharing, còn $P_{new}(u)$ là thành phần phạt khi phải tạo mới instance hoặc kích hoạt thêm node. Các hệ số α , β , γ , δ và η được chọn theo hướng thực nghiệm nhằm bảo đảm thuật toán dễ hiệu chỉnh trong mô phỏng.

Trong phần hiện thực, các thành phần điểm số được chuẩn hóa theo cùng thang đo tương đối để tránh việc một tiêu chí đơn lẻ chi phối toàn bộ quyết định placement. Cụ thể, điểm delay được tính theo tỷ lệ giữa độ trễ đường đi ứng viên và ngưỡng delay của yêu cầu; điểm tài nguyên dựa trên tỷ lệ tài nguyên còn dư của node; điểm reuse và share là các giá trị thưởng nhị phân; còn $P_{new}(u)$ nhận giá trị 1 khi placement làm phát sinh node/instance mới và bằng 0 trong trường hợp ngược lại. Bảng 3.4.1 trình bày bộ trọng số được sử dụng trong các thí nghiệm của luận văn, được trích trực tiếp từ cấu hình script mô phỏng.

Bảng 3.4.1. Trọng số sử dụng trong hàm điểm của DWSA-VNF

Hệ số	Giá trị	Vai trò trong quyết định placement
α	0.35	Ưu tiên phương án có độ trễ đường đi thấp hơn so với ngưỡng delay của yêu cầu
β	0.20	Ưu tiên node còn nhiều tài nguyên CPU và bộ nhớ sau placement
γ	0.20	Thưởng cho việc tái sử dụng node đang hoạt động, hạn chế kích hoạt node mới
δ	0.15	Thưởng cho phương án tận dụng được selective sharing đối với nhóm VNF phù hợp
η	0.10	Phạt phương án phải tạo mới instance hoặc kích hoạt thêm node

Việc chọn α lớn nhất phản ánh mục tiêu delay-aware của phương pháp, vì ràng buộc độ trễ là điều kiện trực tiếp quyết định chất lượng dịch vụ của chuỗi SFC. Hai hệ số β và γ được đặt ở mức trung bình để cân bằng giữa sử dụng tài nguyên còn dư và ưu tiên node đang hoạt động. Hệ số δ khuyến khích reuse có chọn lọc, nhưng không đặt quá cao để tránh thuật toán chia sẻ quá mức. Thành phần phạt η nhỏ hơn các thành phần thưởng chính nhằm tránh việc thuật toán quá bảo thủ trong tạo mới instance khi hệ thống thực sự cần mở rộng placement. Bộ trọng số này không được xem là tối ưu cho mọi môi trường NFV, mà là một cấu hình hợp lý cho phạm vi mô phỏng của luận văn và có thể được hiệu chỉnh trong các nghiên cứu tiếp theo.

Trong phần mô phỏng và đánh giá, các thành phần điểm số này được chuẩn hóa ở mức thực nghiệm để bảo đảm rằng thuật toán giữ được tính ổn định giữa các kịch bản mô phỏng.

Bảng 3.4.2 tóm tắt vai trò của từng thành phần trong cơ chế chấm điểm.

Bảng 3.4.2. Các thành phần của cơ chế chấm điểm trong DWSA-VNF

Thành phần	Ý nghĩa kỹ thuật
S_{delay}	Ưu tiên phương án có lợi hơn về độ trễ
$S_{resource}$	Ưu tiên node còn nhiều tài nguyên khả dụng
S_{reuse}	Ưu tiên sử dụng node đang hoạt động
S_{share}	Cộng điểm nếu tận dụng được selective sharing
P_{new}	Phạt khi phải tạo mới instance hoặc bật node mới

Từ các nguyên tắc trên, quy trình tổng thể của DWSA-VNF có thể được tóm lược như sau. Thuật toán trước hết sắp xếp tập yêu cầu dịch vụ. Sau đó, với từng yêu

cầu, nó xét tuần tự từng VNF trong chuỗi, sinh tập node ứng viên, sàng lọc theo tài nguyên và độ trễ, xét selective sharing, tính điểm và chọn node tốt nhất. Sau mỗi lần placement thành công, trạng thái tài nguyên của hệ thống được cập nhật. Nếu tại bất kỳ bước nào không còn node ứng viên phù hợp, yêu cầu bị từ chối. Nếu toàn bộ chuỗi được ánh xạ thành công, yêu cầu được chấp nhận.

Giả mã tóm tắt của DWSA-VNF được trình bày trong Thuật toán 1.

Algorithm 1 Thuật toán DWSA-VNF

Input: Topology $G = (N, L)$, tập yêu cầu R , tài nguyên node/link, profile workload, ngưỡng delay

Output: Kết quả placement và trạng thái accept/reject cho từng yêu cầu

```

1: Sắp xếp các yêu cầu trong  $R$  theo mức độ ưu tiên
2: for mỗi yêu cầu  $r \in R$  do
3:   Khởi tạo trạng thái tạm thời cho  $r$ 
4:   for mỗi VNF  $f$  trong chuỗi  $S_r$  do
5:     Sinh tập node ứng viên
6:     Loại bỏ node không đủ tài nguyên
7:     Loại bỏ node vi phạm delay
8:     if  $f$  thuộc nhóm chia sẻ được then
9:       Ưu tiên các node có instance cùng loại
10:    else
11:      Chỉ xét tạo mới instance
12:    end if
13:    if không còn node ứng viên then
14:      Đánh dấu  $r$  là reject và dừng xử lý yêu cầu này
15:    end if
16:    Tính điểm cho từng node ứng viên
17:    Chọn node có điểm cao nhất
18:    Cập nhật placement tạm thời
19:  end for
20:  if toàn bộ chuỗi được gán thành công then
21:    Đánh dấu  $r$  là accept và cập nhật tài nguyên chính thức
22:  end if
23: end for

```

Như vậy, DWSA-VNF là một heuristic nhiều tầng, trong đó các nguyên tắc về tài nguyên, delay, workload và selective sharing được chuyển hóa thành một quy trình placement có tính nhất quán. Phương pháp này không theo đuổi mục tiêu tối ưu hóa toàn cục, nhưng đủ chặt chẽ để thể hiện một đóng góp phương pháp rõ ràng cho luận văn, đồng thời đủ gọn để hiện thực hóa trong môi trường mô phỏng.

3.4.1. Độ phức tạp tính toán và khả năng mở rộng

Do DWSA-VNF được thiết kế theo hướng heuristic, việc phân tích độ phức tạp tính toán giúp làm rõ khả năng mở rộng của phương pháp khi số lượng node vật lý và số lượng yêu cầu dịch vụ tăng lên. Gọi $|R|$ là số lượng yêu cầu dịch vụ, $\bar{k}$ là độ dài trung bình của chuỗi VNF, $|N|$ là số node vật lý và C_{path} là chi phí kiểm tra đường đi/ngưỡng delay giữa hai node. Trong trường hợp tổng quát, với mỗi yêu cầu và mỗi VNF trong chuỗi, thuật toán có thể phải xét tối đa $|N|$ node ứng viên. Do đó, độ phức tạp xấp xỉ của phần placement chính có thể viết dưới dạng:

$$O(|R| \times \bar{k} \times |N| \times C_{path}). \quad (3.4.2)$$

Nếu đường đi ngắn nhất giữa các cặp node được tiền xử lý hoặc cache lại, C_{path} có thể giảm đáng kể trong các lần kiểm tra lặp lại. Ngược lại, nếu mỗi lần đánh giá node ứng viên đều chạy lại thuật toán tìm đường đi ngắn nhất trên topology hiện thời, chi phí tính toán sẽ tăng theo số cạnh và số node của đồ thị.

Từ phân tích trên có thể thấy DWSA-VNF không phù hợp để xử lý theo kiểu vét cạn trên các topology rất lớn nếu không có các kỹ thuật tối ưu bổ sung. Tuy nhiên, phương pháp có một số đặc điểm thuận lợi cho mở rộng. Thứ nhất, quá trình sinh node ứng viên có thể giới hạn theo vùng lân cận, theo miền cloud-edge hoặc theo danh sách node có đủ tài nguyên. Thứ hai, thông tin đường đi ngắn nhất và độ trễ giữa các cặp node có thể được tiền xử lý để giảm chi phí lặp lại. Thứ ba, việc chấm điểm các node ứng viên cho cùng một VNF có thể song song hóa vì các điểm số tương đối độc lập trước khi chọn node tốt nhất. Thứ tư, bộ trọng số trong hàm điểm có thể được điều chỉnh theo từng loại môi trường, chẳng hạn cloud tập trung, edge phân tán hoặc multi-domain. Vì vậy, trong phạm vi luận văn, DWSA-VNF được xem là phù hợp cho đánh giá mô phỏng quy mô vừa; với các mạng có hàng nghìn node và số lượng chuỗi dịch vụ rất lớn, cần bổ sung các cơ chế giới hạn ứng viên, tiền xử lý đường đi và/hoặc phân hoạch topology.

Trên cơ sở kết quả placement của DWSA-VNF, Chương 4 sẽ tiến hành đánh giá phương pháp theo các chỉ số chính gồm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối trung bình* (average end-to-end delay), *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization) và *thời gian thực thi* (execution time).

3.5. Môi trường cài đặt và mô phỏng

Để phương pháp DWSA-VNF có thể được kiểm chứng bằng thực nghiệm, luận văn cài đặt toàn bộ tiến trình mô phỏng bằng Python. Lựa chọn này xuất phát từ tính thuận tiện trong triển khai, khả năng xử lý dữ liệu và trực quan hóa kết quả của ngôn ngữ lập trình này. Trong khuôn khổ luận văn, môi trường này được dùng như một nền tảng thống nhất để chạy các baseline và phương pháp đề xuất trên cùng một tập dữ liệu đầu vào.

Chương trình mô phỏng được viết bằng ngôn ngữ phổ biến Python. Điều này có nghĩa là topology vật lý, yêu cầu dịch vụ, workload theo thời gian, các thuật toán placement, các chỉ số đánh giá và phần xuất kết quả đều nằm trong cùng một tiến trình xử lý. Cách làm này giúp giảm thao tác thủ công, tăng tính nhất quán giữa các lần chạy và tạo điều kiện thuận lợi cho việc tái lập thí nghiệm. Về mặt phương pháp luận, đây cũng là một lựa chọn phù hợp vì nó bảo đảm rằng mọi kết quả dùng trong Chương 4 đều được sinh ra từ cùng một pipeline mô phỏng.

Trong môi trường hiện thực này, script trước hết tiếp nhận cấu hình thí nghiệm, sau đó khởi tạo topology và tập yêu cầu dịch vụ. Tiếp theo, script lần lượt chạy các baseline gồm *đặt ngẫu nhiên* (Random Placement), *tham lam ưu tiên nút đầu tiên phù hợp* (Greedy First-Fit) và *tham lam chia sẻ toàn phần* (Greedy Share-All), rồi chạy DWSA-VNF trên cùng một điều kiện đầu vào. Sau mỗi lần chạy, hệ thống ghi lại các chỉ số gồm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối trung bình* (average end-to-end delay), *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization), *số lượng nút hoạt động* (number of active nodes) và *thời gian thực thi* (execution time). Đây cũng chính là bộ chỉ số sẽ được sử dụng nhất quán trong Chương 4.

Một điểm đáng chú ý là script không chỉ sinh ra kết quả cuối cùng, mà còn duy trì được trạng thái tài nguyên của hệ thống trong suốt quá trình placement. Điều này có nghĩa là các quyết định của thuật toán luôn được đánh giá trên trạng thái hạ tầng đã cập nhật, thay vì trên một môi trường tĩnh không đổi. Cách tổ chức như vậy làm cho mô phỏng gần hơn với bản chất động của placement trong NFV, nơi mỗi yêu cầu được chấp nhận sẽ ảnh hưởng tới không gian placement của các yêu cầu tiếp theo.

Toàn bộ đầu ra của script được lưu trong thư mục `thesis_nfv_output`. Trong đó bao gồm các bảng dữ liệu tổng hợp, các tệp cấu hình thí nghiệm và các đồ

thị dùng cho Chương 4. Việc tập trung toàn bộ đầu ra vào cùng một thư mục không chỉ thuận tiện về mặt thao tác, mà còn giúp luận văn bảo đảm tính nhất quán của nguồn dữ liệu sử dụng cho phần phân tích kết quả.

Để tăng khả năng tái lập, luận văn làm rõ thêm một số thành phần cài đặt chính. Script sử dụng Python làm ngôn ngữ hiện thực, kết hợp các thư viện thông dụng như NumPy và Pandas cho xử lý dữ liệu, Matplotlib cho trực quan hóa, cùng các cấu trúc dữ liệu tự cài đặt để biểu diễn node, liên kết, yêu cầu dịch vụ và instance VNF. Thuật toán tìm đường đi ngắn nhất được dùng trong bước kiểm tra delay và lựa chọn tuyến giữa các node placement. Các kết quả trung gian và kết quả tổng hợp được xuất ra các tệp CSV/XLSX và các hình PNG, giúp quá trình kiểm tra lại số liệu và biên soạn chương thực nghiệm được thuận tiện hơn.

Như vậy, phương pháp DWSA-VNF đã được chuyển hóa thành một quy trình có thể chạy được, có thể lặp lại được và có thể tạo ra bộ dữ liệu thực nghiệm phục vụ so sánh với các baseline. Nhờ đó, Chương 4 có thể sử dụng trực tiếp các đầu ra của script để đánh giá hiệu quả của phương pháp đề xuất.

3.6. Kết luận chương

Chương này đã trình bày phần đóng góp phương pháp của luận văn theo hướng thiên về kỹ thuật. Trước hết, luận văn đã xác định cơ sở đề xuất phương pháp từ yêu cầu cải thiện placement/chaining trong môi trường NFV có xét tài nguyên, *độ trễ đầu cuối* (end-to-end delay), workload theo thời gian và selective sharing. Tiếp theo, mô hình hệ thống và các giả thiết mô phỏng đã được xây dựng nhằm tạo khung nhất quán cho quá trình thiết kế thuật toán và mô phỏng.

Trên nền tảng đó, luận văn đã đề xuất phương pháp DWSA-VNF như một heuristic placement có cơ chế sắp xếp yêu cầu, sinh và sàng lọc node ứng viên, kiểm tra tài nguyên, kiểm tra delay, selective sharing và chấm điểm tổng hợp. Mặc dù không theo đuổi mô hình tối ưu hóa toàn cục, DWSA-VNF vẫn thể hiện rõ một logic thiết kế nhất quán và phù hợp với bối cảnh nghiên cứu ứng dụng của luận văn. Cuối cùng, chương đã giới thiệu môi trường cài đặt và tổ chức mô phỏng phương pháp trong chương trình viết bằng Python, qua đó tạo ra nền tảng thực nghiệm thống nhất cho các so sánh ở chương tiếp theo.

Như vậy, Chương 3 đã hoàn thành vai trò là cầu nối giữa phần khảo sát tài liệu và phần đánh giá thực nghiệm. Nếu Chương 2 xác lập bối cảnh nghiên cứu và các hướng tiếp cận liên quan, thì Chương 3 đã cụ thể hóa đóng góp của luận văn bằng một

phương pháp placement có thể mô tả, hiện thực và kiểm chứng. Trên cơ sở mô hình và môi trường mô phỏng đã xây dựng, Chương 4 sẽ tiến hành thiết lập thí nghiệm, so sánh DWSA-VNF với Random Placement, Greedy First-Fit và Greedy Share-All, đồng thời phân tích kết quả theo nhóm chỉ số về khả năng phục vụ, chất lượng dịch vụ và hiệu quả sử dụng tài nguyên.

CHƯƠNG 4. MÔ PHỎNG VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Giới thiệu chương

Kế thừa mô hình hệ thống, phương pháp DWSA-VNF và môi trường mô phỏng đã xây dựng ở Chương 3, chương này tập trung vào việc thiết lập thí nghiệm và đánh giá định lượng hiệu quả của phương pháp đề xuất. Nếu Chương 3 trả lời câu hỏi phương pháp DWSA-VNF được xây dựng như thế nào, thì Chương 4 trả lời câu hỏi phương pháp đó hoạt động ra sao khi được đặt vào một môi trường mô phỏng cụ thể và khi được so sánh với các baseline đại diện. Theo nghĩa đó, chương này đóng vai trò xác nhận thực nghiệm cho phần đóng góp phương pháp của luận văn.

Mục tiêu đánh giá của chương không chỉ dừng ở việc xác định liệu DWSA-VNF có cho kết quả tốt hơn một baseline cụ thể hay không, mà còn nhằm làm rõ cơ chế cải thiện hiệu năng của phương pháp đề xuất dưới nhiều góc độ khác nhau. Trước hết, luận văn cần xem xét khả năng phục vụ dịch vụ của thuật toán thông qua *tỷ lệ chấp nhận yêu cầu* (acceptance ratio) và *thông lượng mạng* (network throughput). Tiếp theo, cần đánh giá liệu việc tăng khả năng phục vụ có làm suy giảm *độ trễ đầu cuối trung bình* (average end-to-end delay) hay không. Sau đó, luận văn cần xem xét mức độ khai thác tài nguyên của hệ thống thông qua *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization), cũng như *số lượng nút hoạt động* (number of active nodes). Cuối cùng, *thời gian thực thi* (execution time) được sử dụng để đánh giá chi phí tính toán tương đối của phương pháp đề xuất khi so với các baseline.

Để bảo đảm tính khách quan của quá trình đánh giá, toàn bộ các thuật toán được triển khai trong cùng một môi trường lập trình Python và được thực thi trên cùng một topology, cùng một tập yêu cầu dịch vụ và cùng một cơ chế sinh workload theo thời gian. Điều này có nghĩa là các khác biệt trong kết quả phản ánh chủ yếu sự khác biệt trong logic placement của từng thuật toán, thay vì xuất phát từ sai khác trong môi trường thực thi. Đây là một nguyên tắc rất quan trọng đối với đánh giá thực nghiệm, bởi chỉ khi các phương pháp được đặt vào cùng một điều kiện đầu vào thì việc so sánh mới có ý nghĩa học thuật và thực tiễn.

Các baseline được sử dụng trong chương này gồm *đặt ngẫu nhiên* (Random

Placement), *tham lam ưu tiên nút đầu tiên phù hợp* (Greedy First-Fit) và *tham lam chia sẻ toàn phần* (Greedy Share-All). Đây là ba đường cơ sở đại diện cho các mức độ khai thác cấu trúc bài toán khác nhau, từ rất đơn giản đến có tận dụng cơ chế dùng chung instance. Random Placement đại diện cho một chiến lược gần như không khai thác cấu trúc bài toán. Greedy First-Fit đại diện cho hướng placement đơn giản nhưng có xét một số điều kiện khả thi cơ bản. Greedy Share-All đại diện cho hướng tham lam nhưng khai thác tối đa việc dùng chung instance. Trên nền so sánh đó, DWSA-VNF được kỳ vọng thể hiện ưu thế nhờ việc kết hợp delay, workload động và selective sharing trong cùng một cơ chế ra quyết định.

Nội dung còn lại của chương được tổ chức như sau. Mục 4.2 trình bày thiết lập môi trường mô phỏng và các kịch bản thí nghiệm. Mục 4.3 mô tả các chỉ số đánh giá được sử dụng trong quá trình so sánh. Mục 4.4 phân tích kết quả theo góc độ khả năng phục vụ và chất lượng dịch vụ. Mục 4.5 phân tích kết quả theo góc độ hiệu quả sử dụng tài nguyên và chi phí vận hành. Cuối cùng, Mục 4.6 đưa ra kết luận chương.

4.2. Thiết lập môi trường mô phỏng và kịch bản thí nghiệm

Để bảo đảm tính nhất quán giữa phần phương pháp và phần đánh giá, các mô phỏng trong luận văn được thực hiện trên đúng môi trường mô phỏng đã mô tả ở Chương 3. Cụ thể, toàn bộ thuật toán được triển khai bằng ngôn ngữ lập trình Python, với script mô phỏng thực hiện lần lượt các bước khởi tạo topology, sinh yêu cầu dịch vụ, gán workload theo thời gian, chạy các baseline, chạy DWSA-VNF, sau đó xuất kết quả sang thư mục `thesis_nfv_output`. Cách tổ chức như vậy giúp bảo đảm rằng mọi dữ liệu dùng trong chương này đều xuất phát từ cùng một pipeline xử lý.

Về mặt cấu hình, môi trường mô phỏng xét ba kích thước topology vật lý gồm 12, 18 và 24 node. Topology được xây dựng theo hướng bảo đảm tính liên thông, trong đó trước hết các node được nối thành một vòng cơ sở, sau đó bổ sung thêm các liên kết ngẫu nhiên để tạo ra nhiều lựa chọn placement và nhiều khả năng kết nối giữa các VNF. Mỗi node vật lý được gán tài nguyên *bộ xử lý trung tâm* (CPU) và bộ nhớ trong khoảng từ 24 đến 36 đơn vị, trong khi băng thông của các liên kết được sinh ngẫu nhiên trong khoảng từ 60 đến 160 đơn vị. Cách thiết lập này giúp môi trường mô phỏng vừa đủ đa dạng để phản ánh các kịch bản placement khác nhau, nhưng vẫn giữ được độ gọn cần thiết cho quá trình thực nghiệm.

Tập yêu cầu dịch vụ được thiết lập theo bốn mức tải lần lượt gồm 30, 60, 90 và 120 yêu cầu. Mỗi yêu cầu được mô hình hóa dưới dạng một *yêu cầu chuỗi chức năng*

dịch vụ (Service Function Chain Request), trong đó độ dài chuỗi VNF thay đổi trong khoảng từ 3 đến 5 chức năng. Bên cạnh đó, mỗi yêu cầu còn được gắn với một profile *tải biến thiên theo thời gian* (time-varying workload) trên 12 *khoảng thời gian rời rạc* (time-slots), cùng với một ngưỡng *độ trễ đầu cuối* (end-to-end delay) được xác định động theo độ dài chuỗi, mức tải trung bình và hệ số siết độ trễ. Trong luận văn này, ba mức hệ số siết độ trễ được sử dụng là 1.0, 0.85 và 0.70, qua đó cho phép đánh giá thuật toán trong các điều kiện SLA từ lỏng đến chặt hơn.

Về cơ chế selective sharing, các VNF thuộc nhóm NAT, Load Balancer và WAN Optimizer được xem là phù hợp với việc dùng chung instance, trong khi Firewall, IDS và DPI được xếp vào nhóm không chia sẻ. Cách phân loại này được giữ nhất quán giữa mô hình hệ thống ở Chương 3, phần hiện thực trong script và các thí nghiệm ở chương này. Để bảo đảm độ tin cậy cho kết quả, mỗi kịch bản thí nghiệm được lặp lại 5 lần và các giá trị trình bày trong chương được tổng hợp từ các lần chạy đó.

Để việc trình bày cấu hình mô phỏng được cô đọng và rõ ràng, Bảng 4.2.1 tóm tắt nhóm tham số chính được sử dụng trong chương. Các giá trị trong Bảng 4.2.1 được trích rút trực tiếp từ cấu hình của script mô phỏng, do đó phản ánh đúng môi trường thực nghiệm sử dụng trong luận văn. Đối với mỗi kịch bản, thí nghiệm được lặp lại 5 lần theo các seed khác nhau và kết quả được tổng hợp để giảm ảnh hưởng của yếu tố ngẫu nhiên trong quá trình sinh topology và sinh yêu cầu dịch vụ.

Bảng 4.2.1. Cấu hình chính của môi trường mô phỏng

Tham số	Giá trị/Cách thiết lập
Số lượng node vật lý	12, 18, 24
Số lượng liên kết vật lý	Topology liên thông dạng ring kết hợp các liên kết ngẫu nhiên bổ sung
Dung lượng CPU mỗi node	24-36 đơn vị
Dung lượng bộ nhớ mỗi node	24-36 đơn vị
Dung lượng băng thông liên kết	60-160 đơn vị
Số lượng yêu cầu dịch vụ	30, 60, 90, 120
Độ dài chuỗi VNF	3-5 VNF/yêu cầu
Số lượng time-slots	12
Ngưỡng độ trễ đầu cuối	1.0, 0.85, 0.70
Số lần lặp mỗi kịch bản	5
Nhóm VNF chia sẻ được	NAT, Load Balancer, WAN Optimizer
Nhóm VNF không chia sẻ	Firewall, IDS, DPI

Bên cạnh các giá trị trung bình, bộ kết quả đầu ra của script còn lưu dữ liệu chi tiết theo từng lần lặp trong các tệp bảng, nhờ đó có thể kiểm tra lại độ biến thiên của kết quả khi cần. Trong bản luận văn này, các hình chính ưu tiên trình bày giá trị trung bình để giữ tính dễ đọc, còn các nhận xét được diễn giải theo xu hướng lặp lại trên các kịch bản khác nhau. Đây là cách trình bày phù hợp với phạm vi mô phỏng của luận văn; trong các nghiên cứu mở rộng, có thể bổ sung thanh sai số, khoảng tin cậy hoặc kiểm định thống kê khi số lần lặp được tăng lên.

Trong chương này, các thuật toán được so sánh gồm bốn phương pháp: *đặt ngẫu nhiên* (Random Placement), *tham lam ưu tiên nút đầu tiên phù hợp* (Greedy First-Fit), *tham lam chia sẻ toàn phần* (Greedy Share-All) và DWSA-VNF. Bảng 4.2.2 tóm tắt ngắn gọn vai trò của từng phương pháp trong ngữ cảnh thí nghiệm. Việc có một bảng so sánh khái quát như vậy giúp người đọc dễ theo dõi hơn khi bước sang các phần phân tích kết quả.

Bảng 4.2.2. Các phương pháp được sử dụng trong so sánh thực nghiệm

Phương pháp	Đặc điểm chính
Random Placement	Chọn placement ngẫu nhiên nếu còn đủ tài nguyên
Greedy First-Fit	Chọn phương án phù hợp đầu tiên theo logic tham lam cơ bản
Greedy Share-All	Greedy có tận dụng dùng chung instance cho mọi VNF cùng loại
DWSA-VNF	Placement có xét độ trễ, workload theo thời gian và selective sharing

Xét về kịch bản mô phỏng, luận văn lựa chọn cách tổ chức theo hướng tách riêng từng góc nhìn đánh giá thay vì dồn toàn bộ vào một thực nghiệm duy nhất. Cách tiếp cận này giúp cho kết quả dễ phân tích hơn, đồng thời cho phép làm rõ từng khía cạnh mà phương pháp đề xuất hướng đến cải thiện. Kịch bản thứ nhất tập trung vào việc thay đổi số lượng yêu cầu dịch vụ, qua đó quan sát khả năng duy trì *tỷ lệ chấp nhận yêu cầu* và *thông lượng mạng* khi tải hệ thống tăng dần. Kịch bản thứ hai tập trung vào ràng buộc *độ trễ đầu cuối*, nhằm xem xét liệu DWSA-VNF có duy trì được chất lượng dịch vụ tốt hơn so với các baseline hay không. Kịch bản thứ ba tập trung vào mức độ khai thác tài nguyên của hệ thống, thông qua các chỉ số như CPU utilization, memory utilization và bandwidth utilization. Kịch bản thứ tư tập trung vào *số lượng nút hoạt động* và *thời gian thực thi*, qua đó phản ánh gián tiếp hiệu quả vận hành và chi phí tính toán của từng thuật toán.

Từ các thiết lập trên có thể thấy rằng, môi trường mô phỏng trong luận văn

vừa đủ đơn giản để triển khai gọn trong một script kịch bản mô phỏng lập trình bằng Python, vừa đủ phong phú để phản ánh được các yếu tố quan trọng của bài toán placement trong NFV. Trên nền tảng đó, phần tiếp theo của chương sẽ làm rõ từng chỉ số đánh giá được sử dụng để so sánh hiệu quả của DWSA-VNF với các baseline.

4.3. Các chỉ số đánh giá

Để đánh giá một cách toàn diện hiệu quả của phương pháp DWSA-VNF, luận văn không chỉ sử dụng một chỉ số đơn lẻ mà sử dụng một nhóm chỉ số phản ánh các khía cạnh khác nhau của bài toán placement trong NFV. Cách tiếp cận này là cần thiết vì hiệu năng của hệ thống không thể được quy về một đại lượng duy nhất. Một thuật toán có thể chấp nhận được nhiều yêu cầu hơn nhưng lại làm tăng độ trễ. Một thuật toán khác có thể giữ độ trễ thấp nhưng lại khai thác tài nguyên kém hiệu quả. Vì vậy, đánh giá định lượng trong chương này được xây dựng theo bốn nhóm chỉ số: khả năng phục vụ, chất lượng dịch vụ, hiệu quả sử dụng tài nguyên và chi phí thực thi.

Chỉ số thứ nhất là *tỷ lệ chấp nhận yêu cầu* (acceptance ratio). Đây là tỷ lệ giữa số lượng yêu cầu dịch vụ được placement thành công và tổng số yêu cầu được đưa vào mô phỏng. Chỉ số này phản ánh trực tiếp năng lực của thuật toán trong việc tiếp nhận và phục vụ dịch vụ trên một hạ tầng hữu hạn tài nguyên. Trong bối cảnh của luận văn, acceptance ratio là một trong những chỉ số quan trọng nhất vì mục tiêu của phương pháp đề xuất là nâng cao khả năng phục vụ của hệ thống mà không làm suy giảm quá mức các tiêu chí hiệu năng khác.

Chỉ số thứ hai là *thông lượng mạng* (network throughput). Nếu acceptance ratio cho biết hệ thống chấp nhận được bao nhiêu yêu cầu, thì throughput cho biết tổng lượng dịch vụ hữu ích mà hệ thống thực sự phục vụ được. Một thuật toán có acceptance ratio cao nhưng chủ yếu chấp nhận các yêu cầu nhỏ có thể vẫn cho throughput thấp hơn một thuật toán khác. Vì vậy, việc xem xét throughput song song với acceptance ratio giúp đánh giá sâu hơn về hiệu quả khai thác của phương pháp placement.

Chỉ số thứ ba là *độ trễ đầu cuối trung bình* (average end-to-end delay). Đây là độ trễ trung bình của các yêu cầu được chấp nhận sau quá trình placement. Chỉ số này phản ánh mức độ mà thuật toán duy trì được *chất lượng dịch vụ* (Quality of Service) cho các chuỗi chức năng dịch vụ. Trong bối cảnh NFV, độ trễ không chỉ phụ thuộc vào bản thân tài nguyên node mà còn phụ thuộc vào cấu trúc đường đi giữa các VNF trong chuỗi. Do đó, average end-to-end delay là chỉ số đặc biệt quan trọng để kiểm tra liệu một thuật toán tăng được acceptance ratio có phải trả giá bằng việc làm chất

lượng dịch vụ giảm xuống hay không.

Nhóm chỉ số tiếp theo phản ánh hiệu quả sử dụng tài nguyên. Trong luận văn, ba chỉ số chính được sử dụng là *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization) và *mức sử dụng băng thông* (bandwidth utilization). Các chỉ số này cho biết thuật toán placement phân bổ tải lên hạ tầng vật lý như thế nào, mức độ tận dụng tài nguyên có cân bằng hay không, và liệu việc cải thiện khả năng phục vụ có đi kèm với việc sử dụng tài nguyên hiệu quả hơn hay không. Đây là nhóm chỉ số có ý nghĩa trực tiếp đối với mục tiêu của DWSA-VNF, bởi phương pháp đề xuất được xây dựng để tăng hiệu quả khai thác hạ tầng thông qua workload-aware placement và selective sharing.

Cuối cùng, *thời gian thực thi* (execution time) được sử dụng để phản ánh chi phí tính toán tương đối của từng thuật toán. Vì DWSA-VNF sử dụng cơ chế sàng lọc và chấm điểm phức tạp hơn so với Random Placement hoặc Greedy First-Fit, thời gian thực thi của phương pháp đề xuất có thể cao hơn các baseline đơn giản. Tuy nhiên, điều này không nhất thiết là một bất lợi nếu chi phí tính toán tăng lên ở mức chấp nhận được và đổi lại là sự cải thiện rõ rệt về khả năng phục vụ hoặc hiệu quả sử dụng tài nguyên. Do đó, execution time được xem như một chỉ số bổ sung để đánh giá sự đánh đổi giữa độ tinh vi của thuật toán và hiệu quả mà nó mang lại.

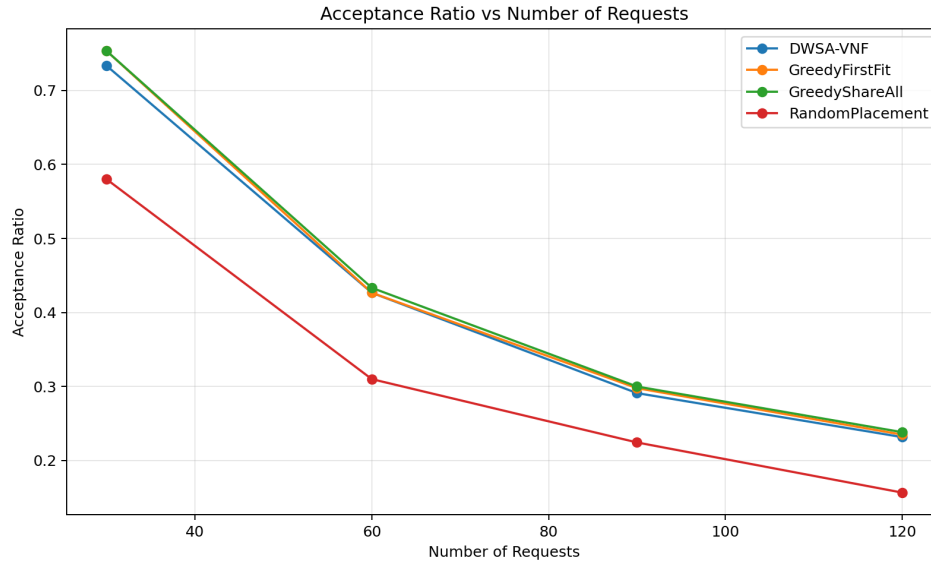
Từ các chỉ số trên, có thể thấy rằng bộ tiêu chí đánh giá trong chương này được xây dựng theo hướng vừa bám sát logic của bài toán placement trong NFV, vừa bám sát các mục tiêu thiết kế của DWSA-VNF ở Chương 3. Nhờ vậy, khi phân tích kết quả ở các mục sau, luận văn không chỉ có thể trả lời câu hỏi phương pháp nào tốt hơn, mà còn có thể trả lời tốt hơn theo nghĩa nào và với sự đánh đổi ra sao.

4.4. Kết quả so sánh theo khả năng phục vụ và hiệu năng dịch vụ

Phần này tập trung phân tích ba chỉ số gắn trực tiếp với khả năng phục vụ và chất lượng dịch vụ của hệ thống, bao gồm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput) và *độ trễ đầu cuối trung bình* (average end-to-end delay). Đây là nhóm chỉ số phản ánh rõ nhất việc DWSA-VNF có thực sự cải thiện được hiệu năng placement so với các baseline hay không.

Trước hết, Hình 4.4.1 trình bày sự so sánh giữa các thuật toán theo chỉ số acceptance ratio. Có thể nhận thấy rằng DWSA-VNF đạt xu hướng tốt hơn so với Random Placement và Greedy First-Fit trong phần lớn các kịch bản. Kết quả này phù hợp với logic thiết kế của phương pháp đề xuất. Trong khi Random Placement gần

như không khai thác cấu trúc của bài toán, còn Greedy First-Fit chủ yếu dựa trên logic tham lam cục bộ, thì DWSA-VNF có khả năng sàng lọc phương án placement dựa trên đồng thời tài nguyên, độ trễ và khả năng selective sharing. Chính sự kết hợp này giúp phương pháp đề xuất duy trì được số lượng yêu cầu được chấp nhận ở mức cao hơn khi áp lực hệ thống gia tăng.

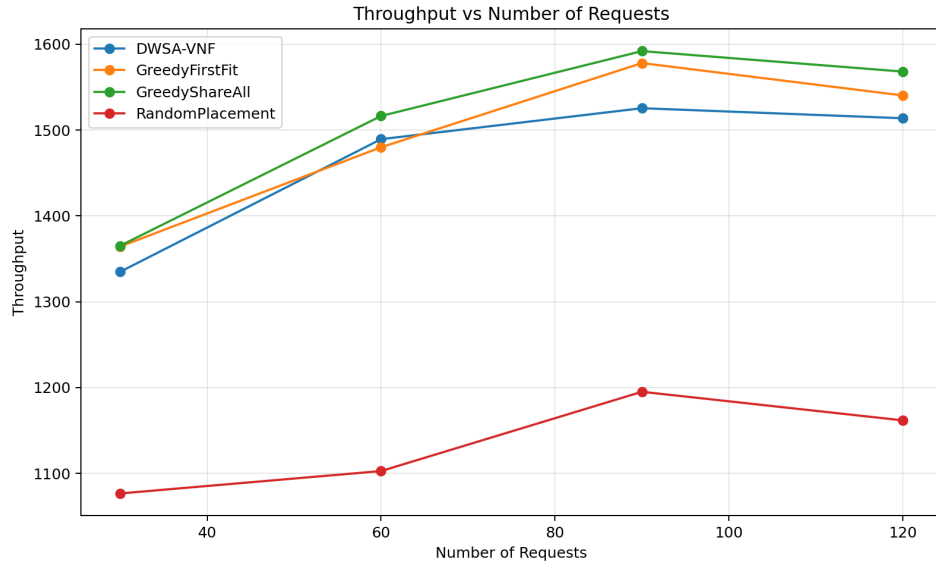


Hình 4.4.1. So sánh tỷ lệ chấp nhận yêu cầu (acceptance ratio) giữa các phương pháp placement

Nếu quan sát sâu hơn dưới góc độ kỹ thuật, có thể lý giải kết quả này như sau. Random Placement thường bỏ lỡ các vùng tài nguyên thuận lợi do không có cơ chế sàng lọc và đánh giá có chủ đích. Greedy First-Fit tuy kiểm tra tính khả thi cơ bản, nhưng thường không tận dụng tốt thông tin về workload động và cũng không phân biệt giữa các trường hợp chia sẻ VNF instance phù hợp hay không phù hợp. Greedy Share-All có khả năng tận dụng reuse mạnh hơn, nhưng vì coi mọi VNF cùng loại đều có thể dùng chung, nó có xu hướng tạo ra một số placement kém hợp lý hơn về mặt mô hình kỹ thuật. Ngược lại, DWSA-VNF chỉ khai thác reuse ở những nhóm VNF được xem là phù hợp, đồng thời còn kết hợp thêm tiêu chí delay và tài nguyên trong hàm điểm. Nhờ đó, acceptance ratio được cải thiện theo hướng bền vững hơn.

Chỉ số thứ hai là *thông lượng mạng* (network throughput), được trình bày trong Hình 4.4.2. Kết quả cho thấy xu hướng của throughput nhìn chung phù hợp với acceptance ratio, nhưng đồng thời cung cấp một góc nhìn sâu hơn về chất lượng phục vụ của các thuật toán. Trong nhiều trường hợp, DWSA-VNF không chỉ chấp nhận được nhiều yêu cầu hơn mà còn phục vụ được tổng lưu lượng cao hơn. Điều này cho thấy

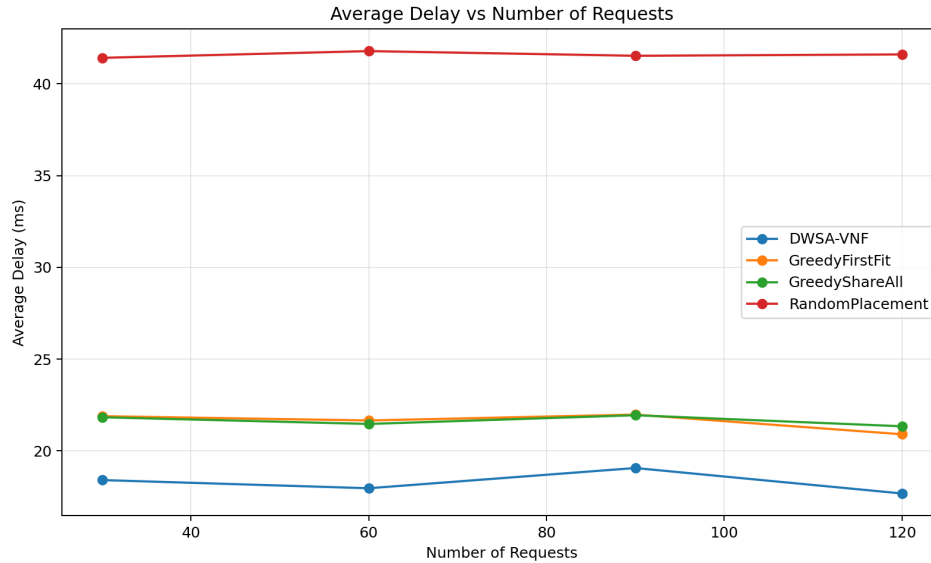
phương pháp đề xuất không chỉ tăng số lượng dịch vụ được nhận vào, mà còn khai thác hạ tầng theo cách giúp tổng tải hữu ích được chuyển hóa thành lưu lượng phục vụ thực tế nhiều hơn.



Hình 4.4.2. So sánh *thông lượng mạng* (network throughput) giữa các phương pháp placement

Ý nghĩa của chỉ số throughput trong bối cảnh này là rất rõ ràng. Nếu một thuật toán đạt acceptance ratio cao nhưng chủ yếu tiếp nhận các yêu cầu nhỏ hoặc placement theo cách làm ngên sớm một số liên kết quan trọng, throughput có thể không tăng tương ứng. Trong trường hợp của DWSA-VNF, việc xét đồng thời delay, tài nguyên và workload động giúp thuật toán tránh được một phần các placement kém hiệu quả như vậy. Kết quả là hệ thống duy trì được tổng lượng dịch vụ phục vụ cao hơn. Đây là một dấu hiệu tích cực cho thấy cải tiến của DWSA-VNF không chỉ mang tính hình thức ở số lượng yêu cầu, mà còn thể hiện ở mức khai thác hạ tầng.

Chỉ số thứ ba là *độ trễ đầu cuối trung bình* (average end-to-end delay), được trình bày trong Hình 4.4.3. Nếu hai chỉ số trước phản ánh khả năng phục vụ, thì chỉ số này phản ánh chất lượng phục vụ. Trong nhiều bài toán placement, một rủi ro thường gặp là thuật toán có thể chấp nhận được nhiều yêu cầu hơn nhưng đồng thời đẩy các chuỗi dịch vụ đi qua các đường dài hơn hoặc placement chúng vào các node kém thuận lợi hơn, làm tăng delay trung bình. Vì vậy, average end-to-end delay là chỉ số đặc biệt quan trọng để đánh giá xem cải thiện acceptance ratio có đi kèm với việc bảo toàn chất lượng dịch vụ hay không.



Hình 4.4.3. So sánh độ trễ đầu cuối trung bình (average end-to-end delay) giữa các phương pháp placement

Kết quả ở Hình 4.4.3 cho thấy DWSA-VNF có khả năng duy trì độ trễ ở mức cạnh tranh, thậm chí tốt hơn so với một số baseline trong nhiều kịch bản. Điều này phù hợp với thiết kế của thuật toán, bởi delay không phải là một chỉ số được đo sau placement, mà là một tiêu chí được đưa vào ngay trong giai đoạn sàng lọc ứng viên và chấm điểm. Nói cách khác, DWSA-VNF không cố gắng tối đa hóa acceptance ratio bằng mọi giá, mà cố gắng đạt được sự cân bằng giữa khả năng chấp nhận dịch vụ và điều kiện SLA. Đây là điểm rất quan trọng từ góc nhìn học thuật, vì nó cho thấy phương pháp đề xuất cải thiện hiệu năng theo hướng thực dụng hơn chứ không chỉ theo hướng gia tăng số lượng dịch vụ được tiếp nhận. Điều này cho thấy việc đưa delay vào giai đoạn sàng lọc và chấm điểm placement là có tác dụng thực tế, chứ không chỉ mang tính hình thức trong thiết kế thuật toán.

Để làm rõ hơn bức tranh tổng thể, Bảng 4.4.1 tóm tắt xu hướng so sánh giữa các thuật toán theo ba chỉ số vừa phân tích. Bảng này kết hợp với các Hình ở trên sẽ đóng vai trò cô đọng hóa nhận xét chính để thuận tiện cho việc so sánh tổng thể.

Bảng 4.4.1. Tóm tắt so sánh theo khả năng phục vụ và chất lượng dịch vụ

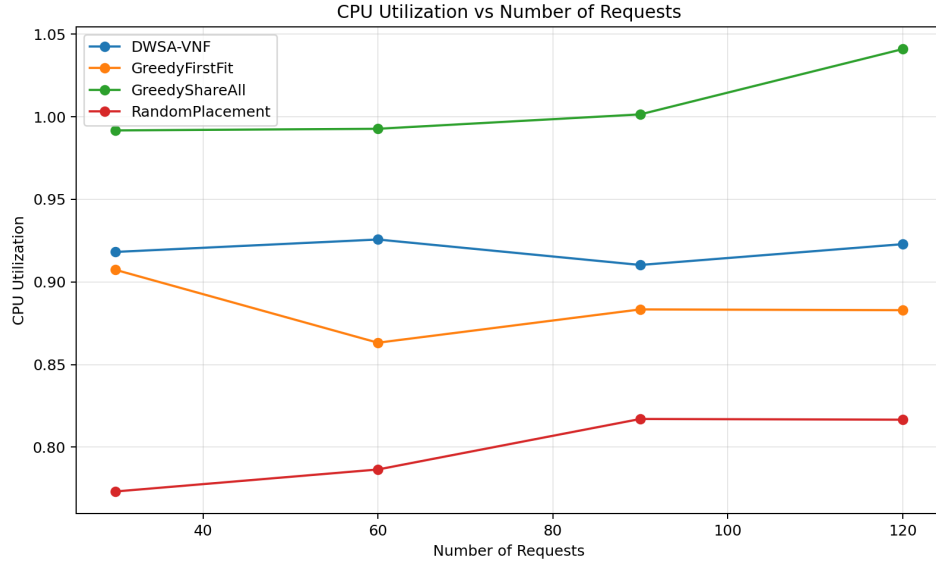
Phương pháp	Acceptance ratio	Throughput	Average delay
Random Placement	Thấp	Thấp	Không ổn định
Greedy First-Fit	Trung bình	Trung bình	Chấp nhận được
Greedy Share-All	Khá	Khá	Có thể dao động theo kịch bản
DWSA-VNF	Tốt	Tốt	Cạnh tranh/ổn định

Từ các kết quả trên, có thể rút ra một số nhận xét quan trọng. Thứ nhất, DWSA-VNF cho thấy xu hướng tốt hơn trong phần lớn kịch bản so với Random Placement và Greedy First-Fit về khả năng chấp nhận và phục vụ dịch vụ. Thứ hai, so với Greedy Share-All, phương pháp đề xuất có xu hướng ổn định hơn do selective sharing giúp tránh một số placement thiếu hợp lý về mặt kỹ thuật. Thứ ba, việc đưa delay vào ngay trong logic thuật toán giúp DWSA-VNF duy trì được chất lượng dịch vụ ở mức tốt hơn thay vì chỉ cải thiện acceptance ratio một cách đơn thuần. Những kết quả này cho thấy phương pháp đề xuất thực sự mang lại giá trị trong nhóm chỉ số phản ánh khả năng phục vụ và hiệu năng dịch vụ.

4.5. Kết quả so sánh theo hiệu quả sử dụng tài nguyên và chi phí vận hành

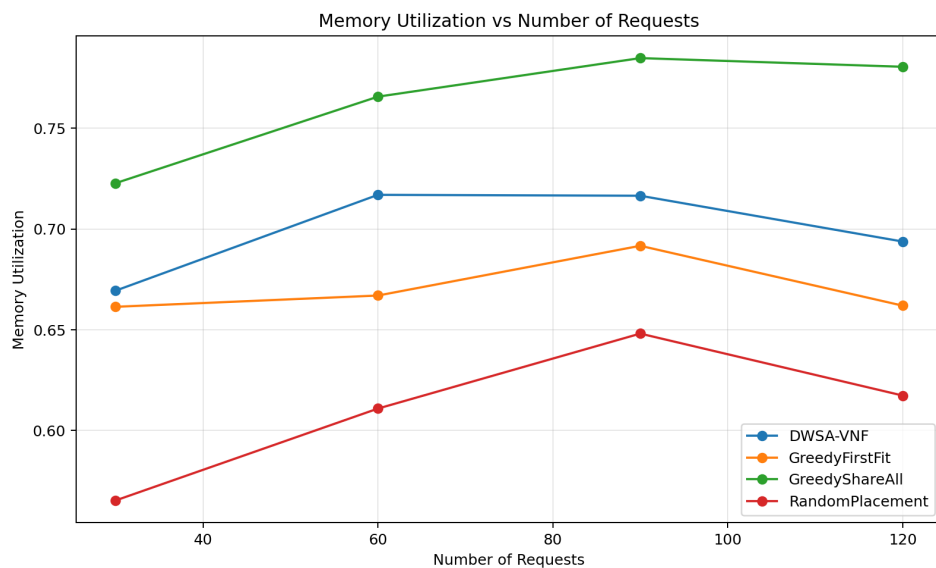
Bên cạnh khả năng phục vụ và chất lượng dịch vụ, một phương pháp placement chỉ thực sự có giá trị nếu nó giúp hệ thống sử dụng tài nguyên hiệu quả hơn. Vì vậy, phần này tập trung phân tích các chỉ số gồm *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization) và *thời gian thực thi* (execution time). Đây là nhóm chỉ số phản ánh chi phí vận hành và mức độ khai thác thực tế của hạ tầng.

Trước hết, Hình 4.5.1 trình bày mức sử dụng CPU giữa các phương pháp placement. Kết quả cho thấy DWSA-VNF có xu hướng khai thác tài nguyên xử lý hiệu quả hơn so với các baseline đơn giản. Điều này phù hợp với logic của thuật toán, bởi DWSA-VNF không chỉ kiểm tra tính khả thi của node, mà còn chấm điểm placement dựa trên mức độ còn dư tài nguyên, trạng thái node đang hoạt động và khả năng chia sẻ VNF instance khi phù hợp. Nhờ đó, thuật toán có xu hướng phân bổ tải theo cách cân bằng và tập trung hơn, thay vì tạo ra quá nhiều điểm placement phân tán.



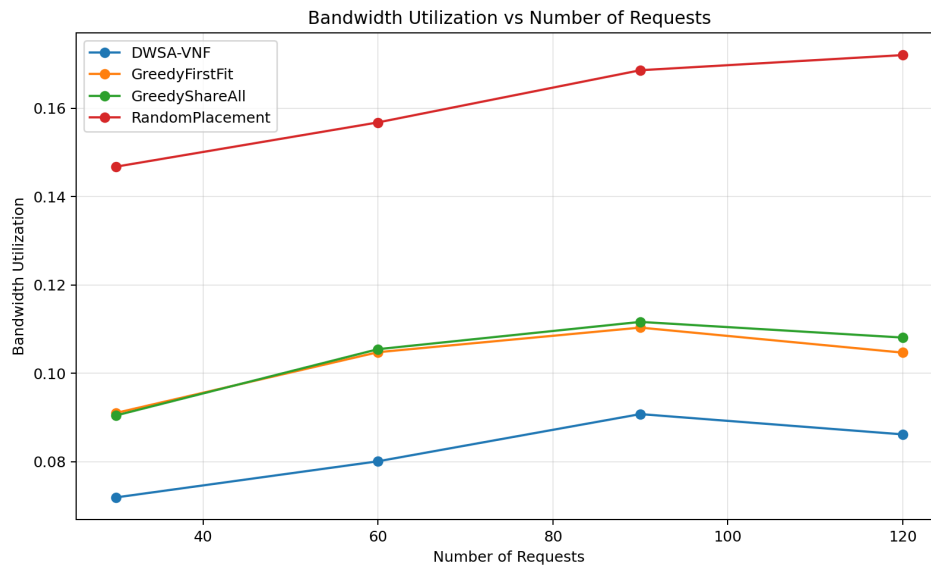
Hình 4.5.1. So sánh mức sử dụng bộ xử lý trung tâm (CPU utilization) giữa các phương pháp placement

Tương tự, Hình 4.5.2 cho thấy sự khác biệt giữa các thuật toán theo mức sử dụng bộ nhớ (memory utilization). Trong môi trường NFV, bộ nhớ có vai trò rất quan trọng vì ngoài phần tài nguyên phục vụ lưu lượng, mỗi VNF instance còn kéo theo một phần chi phí nền. Khi một phương pháp placement tạo ra quá nhiều instance mới, memory utilization có thể tăng nhanh theo hướng thiếu hiệu quả. Kết quả trong mô phỏng cho thấy DWSA-VNF có xu hướng kiểm soát tốt hơn chỉ số này, nhờ selective sharing và ưu tiên tái sử dụng node đang hoạt động.



Hình 4.5.2. So sánh mức sử dụng bộ nhớ (memory utilization) giữa các phương pháp placement

Một chỉ số khác không thể bỏ qua là *mức sử dụng băng thông* (bandwidth utilization), được trình bày trong Hình 4.5.3. Chỉ số này phản ánh mức độ mà thuật toán placement tạo áp lực lên các liên kết vật lý. Trong bài toán NFV, placement không chỉ là câu chuyện của node, mà còn là câu chuyện của đường đi giữa các VNF trong chuỗi. Nếu một phương pháp placement thường xuyên đặt các VNF lên những node quá xa nhau, hoặc không kiểm soát tốt việc định tuyến giữa các bước xử lý, bandwidth utilization trên một số liên kết trọng yếu có thể tăng nhanh và làm giảm hiệu quả chung của hệ thống. Kết quả trong mô phỏng cho thấy DWSA-VNF duy trì được mức sử dụng băng thông ở mức hợp lý, điều này phù hợp với việc delay được xét ngay từ giai đoạn sàng lọc placement.

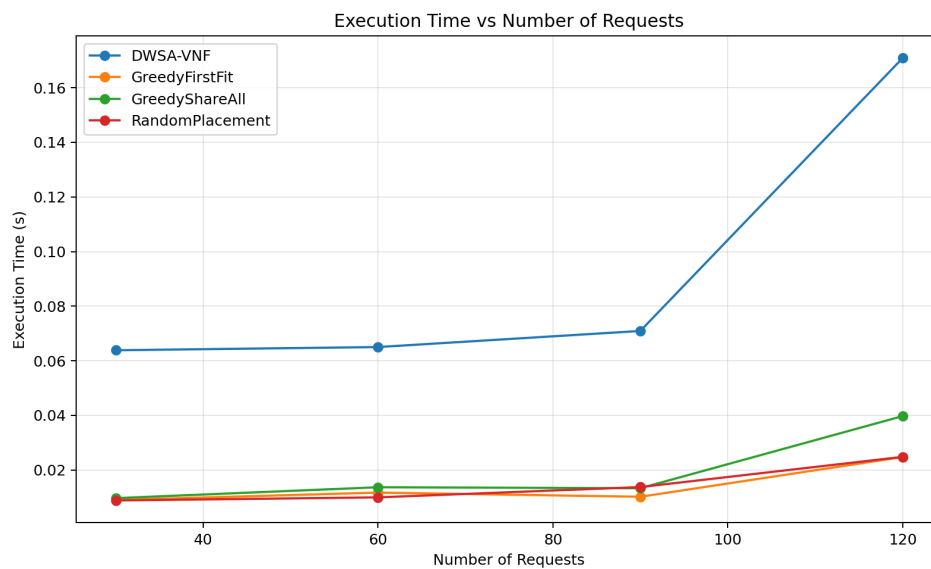


Hình 4.5.3. So sánh mức sử dụng băng thông (bandwidth utilization) giữa các phương pháp placement

Bên cạnh ba chỉ số phản ánh mức độ khai thác tài nguyên, luận văn còn xem xét *thời gian thực thi* (execution time) như một chỉ số bổ sung để đánh giá chi phí tính toán của từng thuật toán. Đây là chỉ số cần thiết vì DWSA-VNF sử dụng nhiều bước xử lý hơn so với Random Placement hoặc Greedy First-Fit, bao gồm sàng lọc node ứng viên, kiểm tra độ trễ và chấm điểm tổng hợp. Do đó, nếu phương pháp đề xuất tạo ra chất lượng placement tốt hơn nhưng chi phí tính toán tăng lên ở mức hợp lý, thì sự đánh đổi này vẫn có thể được xem là chấp nhận được trong bối cảnh nghiên cứu ứng dụng.

Hình 4.5.4 trình bày *thời gian thực thi* (execution time) của các thuật toán. Kết quả cho thấy DWSA-VNF có xu hướng cần nhiều thời gian hơn so với Random

Placement và Greedy First-Fit, điều này là hợp lý vì thuật toán phải thực hiện thêm các bước sàng lọc theo tài nguyên, kiểm tra ràng buộc độ trễ và chấm điểm placement. So với Greedy Share-All, chi phí tính toán của DWSA-VNF cũng có thể cao hơn trong một số kịch bản do selective sharing không chỉ đơn thuần là reuse mọi instance cùng loại, mà còn đòi hỏi kiểm tra tính phù hợp của loại VNF và đánh giá node ứng viên theo nhiều tiêu chí. Tuy nhiên, nếu đặt trong tương quan với những cải thiện về *tỷ lệ chấp nhận yêu cầu, thông lượng mạng* và hiệu quả sử dụng tài nguyên đã phân tích ở các phần trước, mức tăng thời gian thực thi này vẫn là chấp nhận được trong phạm vi mô phỏng của luận văn.



Hình 4.5.4. So sánh thời gian thực thi (execution time) giữa các phương pháp placement

Để tổng hợp các nhận xét chính, Bảng 4.5.1 trình bày so sánh định tính giữa các phương pháp theo nhóm chỉ số tài nguyên và chi phí vận hành.

Bảng 4.5.1. Tóm tắt so sánh theo hiệu quả sử dụng tài nguyên và chi phí vận hành

Phương pháp	CPU/Bộ nhớ	Bảng thông	Execution time
Random Placement	Kém ổn định	Kém ổn định	Thấp
Greedy First-Fit	Trung bình	Trung bình	Thấp
Greedy Share-All	Khá	Khá	Trung bình
DWSA-VNF	Tốt hơn	Ổn định hơn	Trung bình đến khá

Từ nhóm kết quả này có thể rút ra một số kết luận đáng chú ý. Thứ nhất, DWSA-VNF không chỉ cải thiện khả năng phục vụ dịch vụ, mà còn giúp hệ thống khai thác

CPU, bộ nhớ và băng thông hợp lý hơn. Thứ hai, cơ chế workload-aware placement kết hợp với selective sharing giúp phương pháp đề xuất phân bổ tài nguyên theo hướng hiệu quả hơn so với các baseline đơn giản. Thứ ba, mặc dù execution time của DWSA-VNF cao hơn một số baseline, chi phí này là có thể chấp nhận được nếu xét trong mối tương quan với sự cải thiện về các chỉ số quan trọng khác. Nói cách khác, DWSA-VNF thể hiện một sự đánh đổi hợp lý giữa chất lượng quyết định placement và chi phí tính toán.

Nếu kết hợp các nhận xét ở Mục 4.4 và Mục 4.5, có thể thấy rằng hiệu quả của DWSA-VNF không chỉ xuất hiện ở một chỉ số riêng lẻ, mà là ở cả hai nhóm chỉ số chính của bài toán placement NFV. Phương pháp đề xuất vừa cho thấy khả năng nâng cao acceptance ratio và throughput, vừa duy trì được average end-to-end delay ở mức cạnh tranh, đồng thời cải thiện hiệu quả sử dụng tài nguyên và duy trì chi phí tính toán ở mức chấp nhận được. Chính sự đồng thời này tạo nên giá trị thực sự của giải pháp DWSA-VNF.

4.6. Kết luận chương

Chương này đã tiến hành thiết lập thí nghiệm và đánh giá định lượng hiệu quả của phương pháp DWSA-VNF trên cùng một môi trường mô phỏng với các baseline gồm Random Placement, Greedy First-Fit và Greedy Share-All. Quá trình đánh giá được tổ chức theo hai nhóm chỉ số chính, bao gồm nhóm chỉ số phản ánh khả năng phục vụ và chất lượng dịch vụ, cùng với nhóm chỉ số phản ánh hiệu quả sử dụng tài nguyên và chi phí vận hành.

Kết quả mô phỏng cho thấy DWSA-VNF có xu hướng đạt *tỷ lệ chấp nhận yêu cầu* và *thông lượng mạng* tốt hơn so với các baseline đơn giản, đồng thời duy trì *độ trễ đầu cuối trung bình* ở mức cạnh tranh. Điều này cho thấy việc đưa delay vào trong quy trình sàng lọc và chấm điểm placement là hợp lý, giúp phương pháp đề xuất không chỉ tăng số lượng yêu cầu được tiếp nhận mà còn bảo đảm chất lượng phục vụ của các yêu cầu đó. Bên cạnh đó, DWSA-VNF cũng thể hiện hiệu quả tích cực trong khai thác tài nguyên, thông qua các chỉ số *mức sử dụng bộ xử lý trung tâm*, *mức sử dụng bộ nhớ* và *mức sử dụng băng thông*. Mặc dù *thời gian thực thi* của DWSA-VNF cao hơn một số baseline đơn giản, mức chi phí tính toán này vẫn có thể chấp nhận được khi đặt trong tương quan với các cải thiện định lượng mà phương pháp mang lại. Trên cơ sở các kết quả này, phần kết luận chung của luận văn có thể tổng hợp lại đóng góp của phương pháp đề xuất, đồng thời chỉ ra một số hạn chế và hướng phát triển tiếp theo.

Từ góc độ phương pháp luận, kết quả ở chương này cũng xác nhận rằng việc lựa chọn hướng heuristic thiên về kỹ thuật trong Chương 3 là phù hợp với mục tiêu của luận văn. Thay vì xây dựng một mô hình tối ưu hóa quá nặng và khó hiện thực, luận văn đã phát triển một phương pháp vừa đủ đơn giản để triển khai trong lập trình Python, nhưng vẫn đủ mạnh để thể hiện cải thiện rõ rệt trên nhiều chỉ số quan trọng của bài toán placement trong NFV. Điều này giúp bảo đảm rằng đóng góp của luận văn không chỉ nằm ở mức đề xuất ý tưởng, mà đã được kiểm chứng bằng thực nghiệm trong một môi trường mô phỏng thống nhất.

Như vậy, Chương 4 đóng vai trò xác nhận thực nghiệm cho phương pháp DWSA-VNF. Trên cơ sở các kết quả này, phần kết luận chung của luận văn có thể tổng hợp lại những đóng góp chính, nêu ra các hạn chế hiện tại và đề xuất một số hướng phát triển tiếp theo cho nghiên cứu.

KẾT LUẬN

Luận văn với đề tài “*Nghiên cứu công nghệ ảo hóa chức năng mạng (NFV) và đề xuất phương pháp cải tiến hiệu năng*” đã tập trung nghiên cứu một trong những vấn đề quan trọng của môi trường *ảo hóa chức năng mạng* (Network Function Virtualization), đó là bài toán cải thiện hiệu năng thông qua *đặt VNF* (VNF placement) và *xâu chuỗi chức năng dịch vụ* (Service Function Chaining). Trong bối cảnh hạ tầng mạng hiện đại ngày càng mềm hóa, dịch vụ ngày càng đa dạng và yêu cầu vận hành ngày càng linh hoạt, bài toán placement/chaining không chỉ còn là một nội dung kỹ thuật phụ trợ, mà đã trở thành một trong những điểm quyết định trực tiếp tới khả năng khai thác tài nguyên, chất lượng dịch vụ và hiệu quả vận hành của toàn hệ thống.

Trên cơ sở đó, luận văn đã triển khai nghiên cứu theo một quy trình gồm bốn bước chính: xác định bối cảnh và tính cấp thiết của đề tài; khảo sát nền tảng lý thuyết và các hướng nghiên cứu liên quan; đề xuất phương pháp cải tiến hiệu năng; và cuối cùng là kiểm chứng phương pháp bằng mô phỏng thực nghiệm. Cấu trúc này giúp luận văn phát triển tương đối nhất quán từ phần nền tảng khái niệm đến phần đóng góp phương pháp và đánh giá định lượng.

Về mặt tổng quan lý thuyết, luận văn đã hệ thống hóa các nội dung cốt lõi liên quan đến NFV, *mạng điều khiển bằng phần mềm* (Software-Defined Networking), *xâu chuỗi chức năng dịch vụ* (Service Function Chaining), kiến trúc hệ thống NFV, các thành phần của *hạ tầng NFV* (NFV Infrastructure) và *quản lý và điều phối NFV* (Management and Orchestration). Từ quá trình nghiên cứu các tài liệu có liên quan, luận văn cho thấy rằng hiệu năng của hệ thống NFV không chỉ phụ thuộc vào bản thân hạ tầng vật lý, mà phụ thuộc rất mạnh vào cách tài nguyên được phân bổ, cách các *chức năng mạng ảo* (Virtual Network Function) được placement trên hạ tầng, cách dịch vụ được xâu chuỗi và cách lưu lượng được điều hướng qua chuỗi chức năng đó. Luận văn đồng thời đã phân tích các nhóm phương pháp cải tiến hiệu năng chủ yếu, bao gồm *phân bổ tài nguyên* (resource allocation), *điều phối dịch vụ* (service orchestration), placement/chaining, tối ưu hiệu năng thực thi, và các cơ chế bảo đảm khả năng phục hồi hệ thống.

Từ việc phân tích các hướng nghiên cứu hiện có, luận văn xác định được một khoảng trống phù hợp với phạm vi của đề tài. Cụ thể, nhiều công trình hiện nay hoặc

thiên về mô hình tối ưu hóa chính xác với độ phức tạp tính toán lớn, hoặc chưa kết hợp đồng thời các yếu tố quan trọng như *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và cơ chế dùng chung VNF một cách đủ linh hoạt. Trong khi đó, đối với một luận văn thạc sĩ theo hướng ứng dụng, cần một phương pháp vừa có cơ sở kỹ thuật rõ ràng, vừa có thể hiện thực được trong một môi trường mô phỏng thống nhất và đủ gọn nhẹ.

Trên cơ sở khoảng trống nghiên cứu đó, luận văn đã đề xuất phương pháp DWSA-VNF như một heuristic placement/chaining có xét đồng thời ba thành phần chính: *độ trễ đầu cuối* (end-to-end delay), *tải biến thiên theo thời gian* (time-varying workload) và *chia sẻ có chọn lọc* (selective sharing). Điểm cốt lõi của phương pháp là không nhìn placement như một quyết định đơn tiêu chí, mà là một bài toán ra quyết định nhiều yếu tố, trong đó tài nguyên, delay, trạng thái node đang hoạt động, logic dùng chung instance và đặc tính động của workload đều tham gia vào quá trình lựa chọn phương án placement. Thay vì cố gắng xây dựng một mô hình tối ưu hóa quá nặng, luận văn lựa chọn hướng heuristic nhằm đạt được sự cân bằng giữa tính học thuật, tính rõ ràng về kỹ thuật và khả năng triển khai trong thực tế mô phỏng.

Bên cạnh phần phương pháp, một đóng góp quan trọng khác của luận văn là việc xây dựng được một môi trường mô phỏng thống nhất bằng ngôn ngữ lập trình Python. Trong môi trường này, topology vật lý, yêu cầu dịch vụ, workload theo thời gian, các baseline và phương pháp đề xuất đều được xử lý trong cùng một pipeline. Điều này bảo đảm rằng các kết quả thu được có tính nhất quán cao, dễ tái lập và thuận lợi cho việc phân tích, so sánh. Đây là một điểm có ý nghĩa thực tiễn rõ rệt, bởi nó không chỉ phục vụ cho chính luận văn này mà còn có thể làm nền cho các nghiên cứu mở rộng sau đó.

Về mặt thực nghiệm, luận văn đã tiến hành so sánh DWSA-VNF với ba baseline gồm Random Placement, Greedy First-Fit và Greedy Share-All. Việc đánh giá được thực hiện theo nhiều chỉ số định lượng, bao gồm *tỷ lệ chấp nhận yêu cầu* (acceptance ratio), *thông lượng mạng* (network throughput), *độ trễ đầu cuối trung bình* (average end-to-end delay), *mức sử dụng bộ xử lý trung tâm* (CPU utilization), *mức sử dụng bộ nhớ* (memory utilization), *mức sử dụng băng thông* (bandwidth utilization) và *thời gian thực thi* (execution time). Kết quả mô phỏng cho thấy DWSA-VNF có xu hướng cải thiện tốt hơn so với các baseline đơn giản ở các chỉ số về khả năng phục vụ dịch vụ và hiệu quả khai thác tài nguyên, đồng thời vẫn duy trì được độ trễ ở mức cạnh tranh. Mặc dù chi phí tính toán của phương pháp đề xuất cao hơn một số baseline, mức tăng

này vẫn có thể chấp nhận được trong tương quan với những cải thiện định lượng mà phương pháp mang lại.

Từ những kết quả trên, có thể rút ra kết luận rằng hướng tiếp cận của luận văn là phù hợp với mục tiêu đã đặt ra ban đầu. Luận văn không đi theo hướng xây dựng một mô hình tối ưu hóa hình thức quá phức tạp, mà tập trung vào việc lựa chọn một lát cắt tiêu biểu của bài toán NFV placement/chaining để nghiên cứu sâu hơn. Cách tiếp cận này vừa bảo đảm tính khả thi trong phạm vi một luận văn thạc sĩ, vừa vẫn tạo ra được đóng góp có ý nghĩa ở cả phương diện học thuật và thực tiễn. Về mặt học thuật, luận văn góp phần hệ thống hóa tri thức và phát triển một heuristic có cơ sở rõ ràng. Về mặt thực tiễn, luận văn cung cấp một môi trường mô phỏng và một quy trình đánh giá có thể tái sử dụng.

Bên cạnh các kết quả đạt được, luận văn vẫn còn một số hạn chế. Thứ nhất, môi trường đánh giá của luận văn là môi trường mô phỏng, chưa phải là triển khai trên testbed thực tế hoặc trên một nền tảng NFV production-scale như OpenStack, Open Source MANO, Kubernetes, OpenDaylight hoặc ONAP. Do đó, các kết quả thu được chủ yếu phản ánh giá trị của phương pháp trong phạm vi mô hình và cấu hình thực nghiệm đã xác định. Thứ hai, luận văn mới tập trung vào placement/chaining và chưa mở rộng sâu sang các cơ chế như backup, failover, migration, security hardening hay tối ưu hóa mức nền tảng thực thi. Thứ ba, cơ chế selective sharing trong luận văn được xây dựng theo hướng heuristic dựa trên phân loại VNF, chưa đi sâu vào mô hình trạng thái hoặc cô lập dịch vụ ở mức phức tạp hơn. Những hạn chế này không làm mất đi giá trị của kết quả đã đạt được, nhưng là những điểm cần được nhìn nhận thẳng thắn để định vị đúng phạm vi đóng góp của luận văn.

Từ các hạn chế nêu trên, có thể đề xuất một số hướng phát triển tiếp theo. Trước hết, phương pháp DWSA-VNF có thể được mở rộng để xét thêm các cơ chế sao lưu (backup), chuyển đổi dự phòng (failover) hoặc di chuyển (migration) máy ảo/chức năng mạng nhằm tăng mức độ phù hợp với các môi trường NFV yêu cầu tính sẵn sàng cao. Tiếp theo, có thể nghiên cứu mở rộng môi trường placement sang các kiến trúc cloud-edge hoặc multi-domain, nơi delay và tính dị thể của hạ tầng trở nên quan trọng hơn. Một hướng khác là kết hợp heuristic hiện tại với các mô hình học máy hoặc học tăng cường để điều chỉnh động các trọng số hoặc chiến lược placement theo trạng thái hệ thống. Ngoài ra, môi trường mô phỏng của luận văn cũng có thể được mở rộng để hỗ trợ nhiều loại workload hơn, nhiều lớp dịch vụ hơn và các mô hình chia sẻ VNF tinh vi hơn. Về lâu dài, việc kiểm chứng phương pháp trên các testbed thực tế hoặc

các nền tảng NFV mã nguồn mở cũng là một hướng rất đáng quan tâm.

Nhìn chung, luận văn đã cố gắng hoàn thành các mục tiêu nghiên cứu cơ bản đã đặt ra. Từ việc nghiên cứu công nghệ NFV và các hướng cải tiến hiệu năng, luận văn đã đề xuất được một heuristic placement/chaining phù hợp với đặc điểm của môi trường NFV, hiện thực hóa phương pháp đó trong môi trường lập trình, và kiểm chứng hiệu quả thông qua mô phỏng định lượng. Những kết quả đạt được phần nào cho thấy đề tài có tính đúng đắn về định hướng, có ý nghĩa thực tiễn trong nghiên cứu ứng dụng và có thể làm nền cho các hướng phát triển sâu hơn trong tương lai.

DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ

- [CB1] Vũ Việt Dũng, Ngô Hải Anh. *Nghiên cứu công nghệ ảo hóa chức năng mạng (NFV) và phương pháp cải tiến hiệu năng*. Kỷ yếu Hội thảo quốc gia lần thứ XXVIII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, Ninh Bình, 542–548, 15–16/11/2025.

TÀI LIỆU THAM KHẢO

1. Tianzhu Zhang, Han Qiu, Leonardo Linguaglossa, Walter Cerroni, and Paolo Giaccone, *NFV Platforms: Taxonomy, Design Choices and Future Challenges*, IEEE Transactions on Network and Service Management, 2021, 18(1), 30–48.
2. Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba, *Network Function Virtualization: State-of-the-Art and Research Challenges*, IEEE Communications Surveys & Tutorials, 2016, 18(1), 236–262.
3. Haruna Umar Adoga and Dimitrios P. Pezaros, *Network Function Virtualization and Service Function Chaining Frameworks: A Comprehensive Review of Requirements, Objectives, Implementations, and Open Research Challenges*, Future Internet, 2022, 14(2).
4. Zafar Ayyub Qazi, Cheng-Chun Tu, Luis Chiang, Rui Miao, Vyas Sekar, and Minlan Yu, *Simple-fying middlebox policy enforcement using sdn*, SIGCOMM Comput. Commun. Rev., August 2013, 43(4), 27–38.
5. Ahmed M. Medhat, Tarik Taleb, Asma Elmangoush, Giuseppe A. Carella, Stefan Covaci, and Thomas Magedanz, *Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges*, IEEE Communications Magazine, 2017, 55(2), 216–223.
6. Deval Bhamare, Raj Jain, Mohammed Samaka, and Aiman Erbad, *A survey on service function chaining*, Journal of Network and Computer Applications, 2016, 75, 138–155.
7. Juliver Gil Herrera and Juan Felipe Botero, *Resource Allocation in NFV: A Comprehensive Survey*, IEEE Transactions on Network and Service Management, 2016, 13(3), 518–532.
8. Abdelquoddouss Laghrissi and Tarik Taleb, *A Survey on the Placement of Virtual Resources and Virtual Network Functions*, IEEE Communications Surveys & Tutorials, 2019, 21(2), 1409–1434.

9. Song Yang, Fan Li, Stojan Trajanovski, Ramin Yahyapour, and Xiaoming Fu, *Recent advances of resource allocation in network function virtualization*, IEEE Transactions on Parallel and Distributed Systems, 2021, 32(2), 295–314.
10. Khaled Gadouh and Hend Koubaa, *A Literature Survey on Resource Allocation for Network Function Virtualization With and Without Machine Learning in Cloud Computing*, International Journal of Network Management, 2026, 36(1), e70034. e70034 NEM-25-0907.
11. Yi Yue, Shiding Sun, Xuebei Zhang, Ruihua Li, Wencong Yang, Chang Cao, and Xiongyan Tang, *VNF placement in NFV-enabled networks: considering time-varying workloads and multi-tenancy with a throughput optimization heuristic*, Computing, Nov 2024, 106(11), 3657–3690.
12. Defang Li, Peilin Hong, Kaiping Xue, and Jianing Pei, *Availability Aware VNF Deployment in Datacenter Through Shared Redundancy and Multi-Tenancy*, IEEE Transactions on Network and Service Management, 2019, 16(4), 1651–1664.
13. Mina Asgarian, Kamal Jamshidi, and Ali Bohlooli, *An Efficient Approximation Algorithm for Service Function Chaining Placement in Edge-Cloud Computing Industrial Internet of Things*, IEEE Internet of Things Journal, 04 2024, PP.
14. Tiziana Catena, Vincenzo Eramo, Massimo Panella, and Antonello Rosato, *Distributed LSTM-based cloud resource allocation in Network Function Virtualization Architectures*, Comput. Netw., August 2022, 213(C).
15. Yicen Liu and Junning Zhang, *Service Function Chain Embedding Meets Machine Learning: Deep Reinforcement Learning Approach*, IEEE Transactions on Network and Service Management, 2024, 21(3), 3465–3481.
16. Zichuan Xu, Weifa Liang, Alex Galis, Yu Ma, Qiufen Xia, and Wenzheng Xu, *Throughput optimization for admitting NFV-enabled requests in cloud networks*, Comput. Netw., October 2018, 143(C), 15–29.
17. Michel S. Bonfim, Kelvin L. Dias, and Stenio F. L. Fernandes, *Integrated NFV/SDN Architectures: A Systematic Literature Review*, ACM Comput. Surv., February 2019, 51(6).
18. Yong-Xuan Huang and Jerry Chou, *A Survey of NFV Network Acceleration from ETSI Perspective*, Electronics, 2022, 11(9).

19. Lefteris Mamatas, Stuart Clayman, and Alex Galis, *Information Exchange Management as a Service for Network Function Virtualization Environments*, IEEE Transactions on Network and Service Management, 2016, 13(3), 564–577.
20. Hajar Hantouti, Nabil Benamar, Tarik Taleb, and Abdelquoddous Laghrissi, *Traffic Steering for Service Function Chaining*, IEEE Communications Surveys & Tutorials, 2019, 21(1), 487–507.
21. Jianing Pei, Peilin Hong, Kaiping Xue, and Defang Li, *Efficiently Embedding Service Function Chains with Dynamic Virtual Network Function Placement in Geo-Distributed Cloud System*, IEEE Transactions on Parallel and Distributed Systems, 2019, 30(10), 2179–2192.
22. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, Raouf Boutaba, and Otto Carlos Muniz Bandeira Duarte, *Orchestrating Virtualized Network Functions*, IEEE Transactions on Network and Service Management, 2016, 13(4), 725–739.
23. Sebastian Troia, Marco Savi, Giulia Nava, Ligia Maria Moreira Zorello, Thomas Schneider, and Guido Maier, *Performance characterization and profiling of chained CPU-bound Virtual Network Functions*, Computer Networks, 2023, 231, 109815.
24. Defang Li, Peilin Hong, Kaiping Xue, and jianing Pei, *Virtual Network Function Placement Considering Resource Optimization and SFC Requests in Cloud Datacenter*, IEEE Transactions on Parallel and Distributed Systems, 2018, 29(7), 1664–1677.
25. Defang Li, Peilin Hong, Kaiping Xue, and Jianing Pei, *Virtual network function placement and resource optimization in NFV and edge computing enabled networks*, Comput. Netw., April 2019, 152(C), 12–24.