ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN

NGUYỄN MINH PHÚC

NGHIÊN CỨU MÔ HÌNH QUẢN LÝ MẠNG CỤC BỘ VÀ ĐÁM MÂY SỬ DỤNG CÔNG NGHỆ TÁC TỬ DI ĐỘNG

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2026

ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN

NGUYỄN MINH PHÚC

NGHIÊN CỨU MÔ HÌNH QUẢN LÝ MẠNG CỤC BỘ VÀ ĐÁM MÂY SỬ DỤNG CÔNG NGHỆ TÁC TỬ DI ĐỘNG

Chuyên ngành: Quản lý Hệ thống thông tin

Mã số: 9480205.01QTD

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC

1. PGS. TS. Nguyễn Ái Việt

2. TS. Trần Quý Nam

HÀ NỘI - 2026

LỜI CAM ĐOAN

Tôi xin cam đoan luận án "**Nghiên cứu mô hình quản lý mạng cục bộ và đám mây sử dụng Công nghệ tác tử di động**" là công trình nghiên cứu của cá nhân tôi, được hoàn thành dưới sự hướng dẫn của PGS. TS. Nguyễn Ái Việt và TS Trần Quý Nam. Các kết quả nghiên cứu của tôi cùng với các tác giả khác đã được sự nhất trí của các đồng tác giả khi đưa vào nội dung luận án. Tôi đã trích dẫn đầy đủ các tài liệu tham khảo, công trình nghiên cứu liên quan ở trong nước và quốc tế. Tôi xin cam đoan các số liệu và kết quả trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác.

Hà Nội, ngày .... tháng .... năm 2026

Tác giả luận án

Nghiên cứu sinh

Nguyễn Minh Phúc

LỜI CẢM ƠN

Lời đầu tiên, tác giả xin được bày tỏ sự biết ơn chân thành và sâu sắc nhất đến tập thể giáo viên hướng dẫn PGS.TS. Nguyễn Ái Việt và TS. Trần Quý Nam. Các Thầy đã chỉ bảo ân cần và định hướng cho tác giả trong suốt thời gian thực hiện luận án. Các Thầy không những hướng dẫn kiến thức về chuyên môn, học thuật mà còn chỉ bảo cho tác giả những kinh nghiệm trong cuộc sống thường ngày. Một vinh dự rất lớn cho tác giả có cơ hội được học tập, nghiên cứu dưới sự hướng dẫn tận tâm của các Thầy.

Xin bày tỏ sự biết ơn sâu sắc đến các Thầy, Cô trong Viện Công nghệ Thông tin - ĐHQGHN đã luôn quan tâm giúp đỡ và tạo điều kiện về nhiều mặt, chỉ bảo tận tình trong quá trình tác giả thực hiện luận án.

Đặc biệt, xin gửi lời cảm ơn sâu sắc nhất tới gia đình, bạn bè và người thân, những người luôn động viên, chia sẻ và tạo điều kiện tốt nhất cho tác giả có thể học tập, nghiên cứu và hoàn thiện luận án này.

Hà Nội, ngày.. .. tháng .... năm 2026

Tác giả luận án

Nguyễn Minh Phúc

MỤC LỤC

DANH MỤC VIẾT TẮT

TT	Viết tắt	Tiếng Anh	Tiếng Việt
1	ACSE	Association Control Service Element	Thành phần Dịch vụ Điều khiển Liên kết
2	AES	Advanced Encryption Standard	Chuẩn mã hoá nâng cao
3	API	Application Program Interface	Giao tiếp lập trình ứng dụng
4	ASN.l	Abstract Syntax Notation 1	Ký pháp Cú pháp Trừu tượng 1
5	AWS	Amazon Web Service	Dịch vụ điện toán đám mây của Amazon
6	BER	Basic Encoding Rules	Quy tắc mã hoá cơ sở
7	CCITT	Comite Consultatif International de TeIegraphique et TeIephonique	Ủy ban Tư vấn Quốc tế về Điện thoại và Điện báo
8		Cloud Computing	Điện toán đám mây
9	CMIP	Common Management Information Protocol	Giao thức thông tin quản lý chung
10	CMIS	Common Management Information Services	Các dịch vụ thông tin quản lý chung
11	CMISE	Common Management Information Service Element	Thành phần dịch vụ thông tin quản lý chung
12	CMOT	CMIP Over TCP/IP	CMIP qua giao thức TCP/IP
13	DDoS	Distributed Denial of Service	Tấn công từ chối dịch vụ phân tán
14	DME	Distributed Management Environment	Môi trường quản lý phân tán
15	DoS	Denial of Service	Tấn công từ chối dịch vụ
16	GDMO	Guidelines for the Definition of Managed Objects	Hướng dẫn Định nghĩa các Đối tượng Quản lý

TT	Viết tắt	Tiếng Anh	Tiếng Việt
17	GDPR	General Data Protection Regulation	Quy định Bảo vệ Dữ liệu Chung). Đây là một quy định của Liên minh Châu Âu (EU) được áp dụng từ ngày 25 tháng 5 năm 2018, nhằm bảo vệ quyền riêng tư và dữ liệu cá nhân của công dân EU.
18	HIPAA	Health Insurance Portability and Accountability Act	Đạo luật Trách nhiệm và Tính di động trong Bảo hiểm Y tế
19	IAB	Internet Architecture Board	Ban Kiến trúc Internet
20	IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
21	IETF	Internet Engineering Task Force	Lực lượng Đặc nhiệm Kỹ thuật Internet
22	IoT	Internet of Things	Internet vạn vật
23	IP	Internet Protocol	Giao thức Internet
24	IPS	Intrusion Prevention System	Hệ thống Ngăn chặn xâm nhập
25	IRTF	Internet Research Task Force	Lực lượng Đặc nhiệm Nghiên cứu Internet
26	ISO	International Standards Organization	Tổ chức tiêu chuẩn quốc tế
27	ITU	International Telecommunication Union	Liên minh Viễn thông Quốc tế
28	LAN	Local Area Network	Mạng cục bộ
29	LPP	Lightweight Presentation Protocol	Giao thức truyền tải thông tin trình diễn nhẹ. Giao thức này thường dùng trong các hệ thống nhúng

TT	Viết tắt	Tiếng Anh	Tiếng Việt
			hoặc mạng IoT, nơi yêu cầu tiết kiệm băng thông và tài nguyên.
30	MA	Mobile Agent	Tác tử di động
31	MIB	Management Information Base	Cơ sở thông tin quản lý
32	MIT	Management Information Tree	Cây thông tin quản lý
33	MRB	Management Request Broker	Trung gian Yêu cầu Quản lý
34	NCS	Network Control Station	Trạm quản lý mạng
35	NIST	National Institute of Standards and Technology	Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ
36		Network Agent	Tác tử mạng
37	NMA	Network Management Application	Ứng dụng quản trị mạng
38	NME	Network Management Entity	Thực thể quản trị mạng
39	NMS	Network Management System	Hệ thống quản trị mạng
40	OSF	Open Software Foundation	Tổ chức Phần mềm Mở.
41	OSI	Open Systems Interconnection	Kết nối Hệ thống mở
42	PBX	Private Branch eXchange	Tổng đài Nội bộ
43	PCI	Presentation Context Identifier	Định danh ngữ cảnh trình bày
44	PCI DSS	Payment Card Industry Data Security Standard	Tiêu chuẩn Bảo mật Dữ liệu Ngành Công nghiệp Thẻ Thanh toán
45	PDU	Protocol Data Unit	Đơn vị dữ liệu giao thức. PDU là đơn vị dữ liệu được truyền tải giữa các lớp của mô hình OSI, và mỗi lớp có một loại PDU riêng.

TT	Viết tắt	Tiếng Anh	Tiếng Việt
46	PKI	Public Key Infrastructure	Hạ tầng chứng thư số
47	RBAC	Role-Based Access Control	Quản lý quyền truy cập dựa trên vai trò
48	RFC	Request For Comment	Đây là một loạt tài liệu kỹ thuật và tiêu chuẩn được xuất bản bởi IETF (Internet Engineering Task Force) và các tổ chức liên quan, mô tả các giao thức, quy trình, và công nghệ được sử dụng trên Internet.
49	ROSE	Remote Operations Service Element	Thành phần Dịch vụ Thao tác Từ xa. Đây là một giao thức được sử dụng trong mô hình OSI (Open Systems Interconnection) để hỗ trợ các thao tác từ xa giữa các hệ thống.
50	RSA	Rivest-Shamir-Adleman	Thuật toán mã hoá RSA (tên viết tắt của 3 nhà khoa học cho thuật toán này)
51	SDN	Software Defined Network	Mạng Định nghĩa bằng phần mềm
52	SMI	Structure of Management Information	Cấu trúc Thông tin Quản lý. Đây là một phần của SNMP (Simple Network Management Protocol), định nghĩa cách thức tổ chức và biểu diễn thông tin quản lý trong các hệ thống mạng.
53	SNMP	Simple Network Management Protocol	Giao thức quản lý mạng đơn giản

TT	Viết tắt	Tiếng Anh	Tiếng Việt
54	TCP	Transmission Control Protocol	Giao thức Điều khiển Truyền tải. Đây là một trong những giao thức cốt lõi của bộ giao thức TCP/IP, được sử dụng để truyền tải dữ liệu một cách đáng tin cậy giữa các thiết bị trên mạng.
55	TLS	Transport Layer Security	Bảo mật tầng truyền vận
56	UDP	User Datagram Protocol	Giao thức Gói dữ liệu Người dùng
57	VM	Virtual Machine	Máy chủ ảo
58	VPN	Virtual Private Network	Mạng riêng ảo
59	WAN	Wide Area Network	Mạng diện rộng

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

CHƯƠNG MỞ ĐẦU

1. Lý do chọn đề tài

Đề tài nghiên cứu được phát triển từ các chuyên đề nghiên cứu tác tử di động của Nghiên cứu sinh, đồng thời được định hướng bởi bối cảnh ứng dụng của các hệ thống quản lý nội dung và truy cập từ xa trong môi trường mạng cục bộ và Internet, trong đó yêu cầu trọng tâm là bảo đảm truy cập an toàn giữa mạng cục bộ (LAN - Local Area Network) và Internet [6], [7]. Mục tiêu của bối cảnh ứng dụng này là hình thành bộ giải pháp cho phép người dùng truy cập Internet từ mạng cục bộ và truy cập các dịch vụ nội bộ từ Internet một cách an toàn; vì vậy, quản trị mạng an toàn là một cấu phần quan trọng của hệ thống. Trên cơ sở đó, các chuyên đề về tác tử di động tập trung nghiên cứu việc sử dụng tác tử di động để hỗ trợ quản trị mạng an toàn, đáp ứng nhu cầu của quản trị viên trong giám sát, hỗ trợ kỹ thuật và khắc phục sự cố từ xa. Để hạn chế nguy cơ bị lợi dụng quyền truy cập dẫn đến xâm nhập trái phép, công nghệ tác tử di động được đề xuất tích hợp trong một kiến trúc an toàn, an ninh mạng [5].

Các kết quả nghiên cứu của các chuyên đề liên quan [3–6] có thể tiếp tục mở rộng để áp dụng cho nhiều loại mạng khác nhau như mạng di động, mạng định nghĩa bởi phần mềm (SDN – Software Defined Network), mạng 5G và IoT (Internet of Things) những lĩnh vực đang có nhu cầu thực tiễn lớn trong nước và quốc tế. Đặc biệt, trong quá trình dịch chuyển hạ tầng từ mạng cục bộ lên điện toán đám mây (Cloud Computing) của các doanh nghiệp, quản trị mạng trở thành vấn đề then chốt và đòi hỏi lời giải cho các thách thức an ninh mạng phức tạp trong giai đoạn quá độ.

Trong bối cảnh kinh tế số, nhiều tổ chức và doanh nghiệp đang vận hành hạ tầng công nghệ thông tin trên các hệ thống máy chủ vật lý đặt tại chỗ. Mô hình này, dù quen thuộc, đang bộc lộ những hạn chế cố hữu, tạo ra một nhu cầu cấp thiết về việc chuyển đổi sang một mô hình linh hoạt và hiệu quả hơn là điện toán đám mây. Các tổ chức sở hữu và tự vận hành trung tâm dữ liệu riêng thường phải đối mặt với một loạt các thách thức mang tính hệ thống như: Hạn chế về khả năng mở rộng; gánh nặng

về chi phí vốn và chi phí vận hành; rủi ro về tính liên tục trong kinh doanh do dữ liệu và hệ thống tập trung tại một hoặc một vài địa điểm vật lý, khiến chúng dễ bị tổn thương trước các rủi ro như hỏa hoạn, thiên tai, mất điện kéo dài, hoặc các sự cố an ninh vật lý; hạn chế về sự linh hoạt và tốc độ đổi mới.

Từ những thách thức trên, nhu cầu di trú hệ thống lên đám mây trở thành một quyết định chiến lược nhằm giải quyết đồng thời nhiều bài toán như:

- *Tăng cường sự linh hoạt*: Chuyển sang mô hình đám mây cho phép tổ chức co giãn tài nguyên gần như ngay lập tức theo nhu cầu thực tế, đảm bảo hiệu suất ổn định ngay cả trong những giai đoạn cao điểm.

- *Tối ưu hóa mô hình tài chính*: Thay thế chi phí đầu tư vốn lớn bằng chi phí vận hành linh hoạt. Tổ chức doanh nghiệp chỉ trả tiền cho những gì họ thực sự sử dụng, giúp giải phóng nguồn vốn và tối ưu hóa ngân sách.

- *Nâng cao độ tin cậy và an toàn*: Tận dụng hạ tầng đạt chuẩn quốc tế của các nhà cung cấp đám mây, với khả năng sao lưu và phục hồi dữ liệu ở nhiều khu vực địa lý khác nhau, giảm thiểu tối đa thời gian gián đoạn dịch vụ khi có sự cố.

- *Thúc đẩy đổi mới sáng tạo*: Giải phóng đội ngũ kỹ thuật khỏi gánh nặng quản trị hạ tầng, cho phép họ tập trung vào các nhiệm vụ mang lại giá trị cao hơn như phát triển ứng dụng, phân tích dữ liệu và cải tiến quy trình kinh doanh.

Tóm lại, bài toán di trú từ mạng cục bộ lên đám mây là vấn đề chung của nhiều tổ chức, xuất phát từ nhu cầu giải quyết các hạn chế cố hữu của hạ tầng vật lý để trở nên linh hoạt, hiệu quả và có khả năng cạnh tranh tốt hơn trong kỷ nguyên số. Bên cạnh các vấn đề an ninh mạng của mạng cục hoặc mạng đám mây thuần túy, còn có các vấn đề khác nảy sinh cần phải giải quyết [91].

Xuất phát từ yêu cầu trên về việc xây dựng một giải pháp đồng bộ, luận án xác định giải quyết hai vấn đề trọng tâm là mô hình quản trị và công nghệ truy cập. Vấn đề thứ nhất, trong mô hình quản trị, việc sử dụng cơ sở thông tin quản trị (MIB) đã

trở thành tiêu chuẩn. Các hệ thống quản trị dựa trên MIB để đưa ra lệnh điều khiển phù hợp. Hiện nay, cấu trúc mạng ngày càng hiện đại, phức tạp hơn, có nhiều loại thiết bị đa dạng hơn. Tính năng của mỗi thiết bị cũng phức tạp hơn bao gồm cả các phần mềm, phần cứng, cần phải cấu hình, giám sát, xử lý sự cố. Như vậy, MIB bao gồm nhiều đối tượng, hệ thống quản trí cũng phải thu thập và xử lý một khối lượng số liệu ngày càng lớn [8]. Đặc biệt MIB lại thường xuyên cập nhật, trong khi việc xây dựng lại hệ thống phần mềm quản trị lại không thể thực hiện thường xuyên do tốn kém về chi phí và làm gián đoạn vận hành hệ thống. Điều đó đòi hỏi có một mô hình kiến trúc có khả năng cập nhật MIB thường xuyên mà không cần xây dựng lại hệ thống phần mềm, đảm bảo hệ thống hoạt động không bị gián đoạn. Luận án đề xuất giải pháp cho vấn đề này bằng cách sử dụng các lệnh quản tri đã được chuẩn hoá theo chuẩn của ITU [28, 57, 59, 86]. ITU đã có sẵn các chuẩn ngôn ngữ ASN.1 [98] và GDMO [92, 95] để mô tả các đối tượng quản trị (MO – Managed Object) bằng một ngôn ngữ có thể biên dịch ra các ngôn ngữ lập trình và cấu trúc dữ liệu theo chuẩn trên. Khi đó, những lệnh quản trị đối với các đối tượng được quản lý trong MIB sẽ có những lệnh quản trị tương ứng được tự động sinh ra theo thời gian thực trên cơ sở các khuôn mẫu đã được chuẩn hoá.

Vấn đề thứ hai là việc sử dụng công nghệ bảo đảm an toàn cho ứng dụng quản trị hệ thống. Các quản trị viên hỗ trợ sự cố trong mạng cục bộ hay đám mây từ xa thường phải truy cập vào trong mạng. Do đó, các tài nguyên trong mạng có nguy cơ bị tấn công bằng các hành vi không được phép, nếu quyền truy cập bị lộ lọt. Một kịch bản thường xảy ra là từ một thiết bị, quản trị viên còn cần phải truy cập vào các thiết bị khác để thu thập một số lượng lớn thông tin để khắc phục sự cố. Điều đó kéo theo việc toàn bộ mạng có nguy cơ mất an toàn. Công nghệ tác tử di động [56] cho phép các tác tử có thể được gửi đi trong mạng, để thực hiện các tác vụ quản trị đã được định nghĩa từ trước. Do quản trị viên không truy cập vào mạng, các tác vụ không được phép thực hiện sẽ không bao giờ xảy ra. Việc nghiên cứu và làm chủ công nghệ tác tử là một vấn đề rất quan trọng trong việc xây dựng hệ thống quản trị mạng và hứa hẹn sẽ là bước đột phá trong việc xây dựng mô hình quản trị mạng cho hệ thống mạng và điện toán đám mây hiện nay. Mặt khác, ngoài việc làm chủ công nghệ tác tử

thì môi trường tác tử, khung ứng dụng để triển khai tác tử và sự đảm bảo tính an toàn bảo mật cho hệ thống tác tử là một vấn đề trọng yếu để đưa công nghệ tác tử vào triển khai trong thực tế.

Luận án này giải quyết hai vấn đề nêu trên để xây dựng một giải pháp quản trị mạng cục bộ trong quá trình dịch chuyển lên mạng đám mây an toàn dựa trên công nghệ tác tử. Vấn đề thứ nhất được giải quyết bằng phương pháp kiến trúc để kết hợp các chuẩn quản trị mạng SNMP và CMIP. Vấn đề thứ hai được giải quyết bằng công nghệ tác tử và một giải pháp tăng cường tính an toàn của các công nghệ tác tử nguồn mở.

Bên cạnh, vấn đề phát triển một công nghệ quản trị mạng mới, nghiên cứu cũng đưa ra một số cách tiệm cận mới tới bài toán quản trị mạng cục bộ trong quá trình dịch chuyển lên đám mây. Trước hết, xu hướng quản trị mạng hiện nay đều dựa trên hướng tập trung do phải sử dụng các ứng dụng được cài đặt trên các thiết bị, máy trạm để thu thập thông tin và đưa toàn bộ dữ liệu về để xử lý dẫn tới tình trạng tiêu tốn băng thông mạng, sử dụng nhiều nguồn tài nguyên hệ thống để xử lý và phân tích số liệu và kết xuất báo cáo. Với công nghệ tác tử di động và sự triển khai mô hình quản trị mạng theo một phương thức hoàn toàn mới, hứa hẹn sẽ đem lại sự hiệu quả do đặc tính của công nghệ tác tử đó là sự dịch chuyển mã nguồn từ máy này sang một máy khác mà không ảnh hưởng tới hệ thống, sử dụng ít tài nguyên hệ thống do dữ liệu được xử lý ngay tại máy chứa tác tử và tiêu tốn ít băng thông mạng do không cần truyền lượng lớn dữ liệu giữa các thiết bị trong hệ thống mạng. Tuy nhiên, việc sử dụng công nghệ tác tử vào lĩnh vực quản trị mạng cần phải thực hiện tốt việc xây dựng chuẩn hoá công nghệ tác tử và xây dựng chuẩn quản trị mạng dựa trên nền công nghệ tác tử di động cho các thiết bị mạng mới kết hợp với các chuẩn giao thức quản trị mạng SNMP đang được sử dụng rộng rãi cho các thiết bị cũ sẽ đảm bảo tính tương thích với các thiết bị đang sử dụng hiện nay mà đem lại tính hiệu quả từ công nghệ mới đem lại.

2. Mục đích (hoặc mục tiêu) và nhiệm vụ nghiên cứu

Mục tiêu nghiên cứu của luận án tập trung vào các vấn đề cốt lõi sau:

- Nội dung 1: Nghiên cứu và cải tiến đặc tả quản lý mạng trên nền tảng tác tử di động

Nghiên cứu này tập trung vào ba hướng chính nhằm xây dựng một hệ thống quản trị mạng hiệu quả và an toàn. Đầu tiên, nghiên cứu về hệ thống mạng LAN, ảo hóa hạ tầng mạng trên Cloud và công nghệ mạng SDN (Software Defined Network) nhằm tạo nền tảng cho việc quản lý mạng linh hoạt và hiệu quả. Thứ hai, nghiên cứu về các chuẩn quản lý mạng trong mạng cục bộ và đám mây, bao gồm chuẩn ngôn ngữ giao tiếp ASN.1 (Abstract Syntax Notation One), chuẩn định dạng GDMO (Guidelines for the Definition of Managed Objects), và các phiên bản của giao thức quản lý mạng SNMP (SNMPv1, SNMPv2, SNMPv3) cũng như CMIP/CMISE (Common Management Information Protocol/Common Management Information Service). Cuối cùng, nghiên cứu và cải tiến kỹ thuật để đảm bảo tính an toàn và bảo mật cho hệ thống thông tin dựa trên nền tảng tác tử di động, từ đó nâng cao khả năng chống lại các mối đe dọa bảo mật trong môi trường mạng hiện đại.

- Nội dung 2: Nghiên cứu công nghệ tác tử di động và ứng dụng công nghệ tác tử

Nghiên cứu này tập trung vào việc tìm hiểu công nghệ tác tử di động, bao gồm các đặc điểm và tính chất nổi bật so với các mô hình truyền thống như chủ Khách (client/server), web, và CORBA (Common Object Request Broker Architecture). Công nghệ tác tử di động được phân tích dựa trên kiến trúc nền tảng của nó, bao gồm các khía cạnh như triệu gọi hàm từ xa, ngôn ngữ lập trình dành cho tác tử, quản lý quá trình thực thi tác tử, kết nối giữa các tác tử, xác định vị trí tác tử, và đặc biệt là các giải pháp đảm bảo tính an toàn và bảo mật cho tác tử. Bên cạnh đó, nghiên cứu cũng khám phá các lĩnh vực ứng dụng thực tế của công nghệ tác tử, từ quản lý mạng, tự động hóa hệ thống đến các ứng dụng trong lĩnh vực thương mại điện tử và IoT (Internet of Things), nhằm đánh giá tiềm năng và hiệu quả của công nghệ này trong các tình huống cụ thể.

- Nội dung 3: Nghiên cứu và xây dựng bộ khung ứng dụng trên nền tác tử di động theo chuẩn FIPA

Nghiên cứu này tập trung vào việc tìm hiểu và áp dụng bộ chuẩn tác tử di động FIPA (Foundation for Intelligent Physical Agents), bao gồm các đặc tả quan trọng như kiến trúc tác tử, ngôn ngữ nội dung tác tử (ACL - Agent Communication Language), và quản lý tác tử. Các đặc tả này cung cấp nền tảng để thiết kế và triển khai các hệ thống tác tử di động một cách chuẩn hóa và hiệu quả. Bên cạnh đó, nghiên cứu cũng hướng đến việc xây dựng hoặc cải tiến bộ khung ứng dụng nền tác tử di động, nhằm tối ưu hóa khả năng tương tác, quản lý và bảo mật của các tác tử trong môi trường phân tán và đa dạng. Qua đó, nghiên cứu góp phần nâng cao tính linh hoạt và hiệu quả của các hệ thống dựa trên công nghệ tác tử trong các ứng dụng thực tế.

- Nội dung 4: Xây dựng và nâng cao an toàn thông tin và bảo mật cho hệ thống mạng dựa trên nền tảng công nghệ tác tử di động và đảm bảo an toàn thông tin cho nền tảng tác tử di động.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của luận án tập trung vào các nghiên cứu công nghệ tác tử, các chuẩn quốc tế (ISO, ITI-U) về quản lý mạng, phương thức đảm bảo an toàn bảo mật cho các hệ thống thông tin dựa trên công nghệ tác tử di động:

- Tích hợp các giao thức quản lý mạng SNMP, CMIP và các giao thức quản lý mạng khác.

- Ứng dụng công nghệ tác tử di động vào trong thực tế, và ứng dụng công nghệ tác tử vào trong lĩnh vực quản lý mạng.

- Cải tiến bộ khung ứng dụng trên nền tác tử di động theo chuẩn FIPA.

- Cải tiến kỹ thuật cho đặc tả quản lý mạng trên nền tảng tác tử di động.

- Đề xuất mô hình ứng dụng tác tử và nâng cao an toàn bảo mật cho hệ thống quản lý mạng dựa trên nền tảng công nghệ tác tử di động và đảm bảo an toàn thông tin cho nền tảng tác tử di động.

4. Phương pháp nghiên cứu

- Dựa trên phân tích các tài liệu nghiên cứu và giải pháp liên quan về công nghệ tác tử di động, ứng dụng tác tử di động, bộ khung ứng dụng tác tử di động để chỉ ra các ưu và nhược điểm còn tồn tại để cải tiến kỹ thuật và dựa trên nghiên cứu mô hình

từ các bộ khung ứng dụng tác tử di động để xây dựng, kế thừa và nâng cấp bộ khung ứng dụng tác tử di động.

- Nghiên cứu từ các chuẩn quốc tế về chuẩn về quản lý mạng SNMPv1, SNMPv2, SNMPv3 và chuẩn CMISE/CMIP theo chuẩn OSI, chuẩn về ngôn ngữ giao tiếp giữa các bộ giao thức trừu tượng ASN.1/GDMO và các chuẩn về thông tin quản lý MIBv1, MIBv2, tác tử di động FIPA,... Từ nghiên cứu các chuẩn quốc tế, tiến hành thực nghiệm để cải tiến kỹ thuật, đặc tả kỹ thuật để xây dựng bộ giao thức bảo mật cho hệ thống thông tin dựa trên nền tảng tác tử và quản lý mạng dựa trên nền tảng tác tử.

- Thử nghiệm các kết quả nghiên cứu trên hệ thống Lab giả lập; trong trường hợp điều kiện cho phép, có thể triển khai thử nghiệm giải pháp trên hệ thống thông tin tại đơn vị đang công tác và một số công ty CNTT.

5. Đóng góp của luận án

Nghiên cứu này tập trung vào việc giải quyết hai vấn đề chính nhằm xây dựng một hệ thống quản trị mạng cục bộ và đám mây an toàn, hiệu quả dựa trên nền tảng tác tử di động.

Vấn đề thứ nhất tập trung vào việc đề xuất mô hình quản lý mạng cục bộ và đám mây trên nền tảng tác tử di động. Mô hình này tận dụng tính linh hoạt và khả năng tự động hóa của công nghệ tác tử để quản lý hiệu quả các tài nguyên mạng trong cả môi trường truyền thống và đám mây. bao gồm việc đề xuất giải pháp kết nối giữa hai giao thức quản lý mạng SNMP (Simple Network Management Protocol) và CMIP (Common Management Information Protocol). Giải pháp này nhằm tận dụng ưu điểm của cả hai giao thức: SNMP với tính đơn giản và phổ biến, cùng CMIP với khả năng quản lý phức tạp và tính bảo mật cao. Bên cạnh đó, những cải tiến này nhằm đảm bảo tính ổn định, an toàn và hiệu quả của hệ thống quản trị mạng dựa trên tác tử di động trong bối cảnh công nghệ hiện đại.

Vấn đề thứ hai: nghiên cứu đề xuất các cải tiến kỹ thuật cho nền tảng tác tử di động, bao gồm tối ưu hóa quy trình thực thi, nâng cao khả năng tương tác giữa các tác tử, và cải thiện các cơ chế bảo mật như mã hóa dữ liệu, xác thực danh tính và quản

lý quyền truy cập. Đồng thời, nghiên cứu cũng đề xuất một mô hình phát hiện xâm nhập (Intrusion Detection System - IDS) tích hợp trên nền tảng SDN (Software-Defined Networking) để nâng cao tính an toàn và bảo mật cho hệ thống mạng cục bộ và đám mây. Mô hình này kết hợp các kỹ thuật giám sát, phân tích hành vi mạng và phản ứng tự động nhằm ngăn chặn các mối đe dọa tiềm ẩn.

6. Bố cục luận án

Luận án gồm các phần sau:

Phần mở đầu trình bày lý do chọn đề tài; Giới thiệu mục tiêu, đối tượng, phạm vi và phương pháp nghiên cứu; Ý nghĩa khoa học của đề tài; Trình bày bố cục luận án.

Chương 1 trình bày tổng quan về quản lý mạng, bài toán quản lý mạng cục bộ và đám mây, nền tảng tác tử di động và ứng dụng Công nghệ tác tử di động và các giao thức trong quản lý mạng, ứng dụng của tác tử di động vào quản lý hệ thống mạng.

Chương 2 trình bày đề xuất Mô hình kiến trúc Quản lý mạng cục bộ và đám mây trên nền tảng tác di động giúp cho việc quản trị mạng mạng cục bộ và đám mây hiệu quả và tối ưu.

Chương 3 trình bày về kỹ thuật ứng dụng tác tử di động để nâng cao an toàn bảo mật thông tin cho hệ thống mạng cục bộ và đám mây, đồng thời đề xuất cải tiến và nâng cao an toàn bảo mật cho nền tác tử di động.

Phần kết luận nêu những đóng góp chính của luận án, các hướng phát triển nghiên cứu tiếp theo và những vấn đề quan tâm của tác giả; danh mục các công trình đã được công bố của liên quan tới nội dung luận án; danh sách tài liệu tham khảo được sử dụng trong luận án.

CHƯƠNG 1 TỔNG QUAN VỀ QUẢN LÝ MẠNG CỤC BỘ VÀ ĐÁM MÂY

1.1. Bài toán dịch chuyển mạng cục bộ (LAN) lên đám mây (Cloud)

Bài toán dịch chuyển mạng cục bộ lên đám mây phức tạp và yêu cầu giải pháp toàn diện. Trong đó, các nội dung cần giải quyết là vấn đề bảo mật dữ liệu và hệ thống, đảm bảo hiệu suất và độ trễ, khả năng tích hợp hệ thống và tương thích, khả năng quản lý tài nguyên giám sát hoạt động và tuân thủ pháp lý, chi phí vận hành.

Luận án tập trung vào nghiên cứu hai nội dung chính là đảm bảo an toàn thông tin cho hệ thống thông tin và quản lý mạng hiệu quả.

1.1.1 Vấn đề đảm bảo an toàn thông tin trong dịch chuyển mạng cục bộ lên đám mây

Dịch chuyển từ mạng cục bộ từ môi trường truyền thống lên nền tảng điện toán đám mây mang lại nhiều lợi ích, như khả năng mở rộng, tính linh hoạt và tiết kiệm chi phí. Tuy nhiên, quá trình này cũng đi kèm với các thách thức an toàn đáng kể, đòi hỏi các tổ chức phải đối mặt với các mối đe dọa mới và triển khai các biện pháp bảo mật hiệu quả [29].

Một số vấn đề chính bao gồm bảo mật dữ liệu, bảo vệ hệ thống, duy trì quyền riêng tư, kiểm soát truy cập và đảm bảo tính sẵn sàng của dịch vụ. Việc chuyển đổi không chỉ làm thay đổi kiến trúc hệ thống, mà còn đòi hỏi sự phối hợp giữa các chính sách bảo mật truyền thống và các giải pháp bảo mật đám mây hiện đại [71].

Nguy cơ đe dọa an toàn khi dịch chuyển lên đám mây

- Rủi ro mất an toàn dữ liệu [46]

+ Rò rỉ dữ liệu: Khi dữ liệu được lưu trữ trên đám mây, nguy cơ bị đánh cắp hoặc rò rỉ thông tin từ các cuộc tấn công mạng hoặc các nhà cung cấp dịch vụ đám mây không đáng tin cậy gia tăng.

+ Mất mát dữ liệu: Lỗi kỹ thuật, tấn công mã độc hoặc các vấn đề khác có thể dẫn đến mất mát dữ liệu quan trọng, đặc biệt nếu không có các biện pháp sao lưu và phục hồi thích hợp.

- Tấn công vào cơ sở hạ tầng đám mây [32]

+ Tấn công từ chối dịch vụ (DoS – Denial of Service): Các cuộc tấn công DoS hoặc DDoS (Distributed Denial of Service) có thể làm gián đoạn tính sẵn sàng của các dịch vụ đám mây, ảnh hưởng đến hoạt động của tổ chức.

+ Tấn công xen giữa (Man-in-the-Middle): Kẻ tấn công có thể chặn dữ liệu khi nó di chuyển giữa mạng cục bộ và môi trường đám mây, gây nguy cơ giả mạo hoặc đánh cắp thông tin nhạy cảm.

- Vấn đề kiểm soát truy cập [39]

+ Quyền truy cập không phù hợp: Khi dịch chuyển lên đám mây, việc phân quyền không chính xác có thể dẫn đến rủi ro khi người dùng không có thẩm quyền truy cập vào dữ liệu hoặc hệ thống quan trọng.

+ Lạm dụng đặc quyền: Những người dùng có quyền cao (như quản trị viên) có thể sử dụng quyền của mình để thực hiện các hành vi không được phép.

- Vấn đề tuân thủ và quyền riêng tư [35]

+ Quy định pháp lý: Lưu trữ dữ liệu trên đám mây có thể khiến tổ chức gặp khó khăn trong việc tuân thủ các quy định pháp lý liên quan đến quyền riêng tư, như GDPR hoặc HIPAA.

+ Địa điểm dữ liệu: Các nhà cung cấp đám mây lưu trữ dữ liệu tại nhiều địa điểm khác nhau, làm gia tăng nguy cơ vi phạm các quy định về lưu trữ dữ liệu.

Yêu cầu xây dựng Kiến trúc và chuẩn hoá an toàn mạng trong thiết kế hệ thống

Kiến trúc an ninh mạng đóng vai trò như một nền tảng vững chắc để xây dựng và triển khai các hệ thống bảo mật. Nó giúp xác định các điểm yếu tiềm ẩn, xây dựng các lớp phòng thủ và đảm bảo rằng mọi thành phần của hệ thống được bảo vệ trước các mối đe dọa. [42]

Một kiến trúc an ninh mạng được thiết kế tốt giúp phân bổ nguồn lực hiệu quả, tránh trùng lặp hoặc thiếu hụt trong triển khai các biện pháp an ninh. Điều này đặc biệt quan trọng trong các hệ thống lớn, phức tạp hoặc yêu cầu hiệu năng cao [79].

Hệ thống an ninh mạng dựa trên kiến trúc tốt có khả năng mở rộng và thích ứng với các thay đổi trong nhu cầu kinh doanh, công nghệ hoặc môi trường đe dọa.

Vai trò của các chuẩn an ninh mạng trong thiết kế hệ thống như sau:

- Đảm bảo tính tương thích và tương tác: Chuẩn an ninh mạng đảm bảo rằng các thành phần của hệ thống, dù đến từ các nhà cung cấp khác nhau, có thể hoạt động cùng nhau một cách trơn tru. Ví dụ, giao thức TLS (Transport Layer Security) được tiêu chuẩn hóa để mã hóa thông tin truyền tải qua mạng [31].

- Định hướng thiết kế và triển khai: Các chuẩn an ninh mạng cung cấp các hướng dẫn và quy tắc rõ ràng, giúp đơn giản hóa quá trình thiết kế và triển khai hệ thống. Các tổ chức như ISO và NIST đưa ra các tiêu chuẩn như ISO/IEC 27001 hoặc NIST Cybersecurity Framework, giúp định hình các chính sách và biện pháp an ninh [59], nhóm VNITA cũng đưa ra khung đánh giá an toàn thông tin cho hệ thống thông tin [68].

- Đáp ứng yêu cầu pháp lý và tuân thủ: Tuân thủ các chuẩn an ninh mạng giúp doanh nghiệp đáp ứng các yêu cầu pháp lý và chuẩn mực ngành. Ví dụ, tiêu chuẩn PCI DSS là bắt buộc đối với các tổ chức xử lý thẻ thanh toán [78].

- Tăng cường lòng tin từ khách hàng và đối tác: Việc áp dụng các chuẩn an ninh mạng giúp doanh nghiệp xây dựng lòng tin với khách hàng và đối tác, vì họ có thể đảm bảo rằng các dữ liệu nhạy cảm được bảo vệ đúng cách.

Tại sao cần kiến trúc chuẩn đảm bảo an ninh mạng?

Việc thiếu một kiến trúc chuẩn sẽ dẫn đến các vấn đề nghiêm trọng:

- Cấu hình sai, tạo lỗ hổng: Mỗi kỹ sư, mỗi đội nhóm có thể tự ý cấu hình các dịch vụ bảo mật theo cách riêng, dẫn đến sự thiếu nhất quán và dễ dàng tạo ra các lỗ hổng chết người (ví dụ: mở nhầm cổng tường lửa, đặt sai quyền truy cập kho dữ liệu).

- Khó khăn trong quản lý và tuân thủ: Khi hệ thống phình to, việc kiểm tra xem toàn bộ hệ thống có đang tuân thủ các quy định pháp lý (ví dụ: Luật An ninh mạng số 24/2018/QH14) và tiêu chuẩn ngành (ví dụ: PCI-DSS cho thanh toán thẻ) trở nên bất khả thi nếu không có một thiết kế gốc để đối chiếu.

- Lãng phí và kém hiệu quả: Việc lặp đi lặp lại công việc thiết kế các biện pháp bảo mật cho từng ứng dụng mới gây tốn kém thời gian, nguồn lực và dễ phát sinh lỗi.

Vì vậy, một kiến trúc an ninh mạng chuẩn đóng vai trò như một bản thiết kế tổng thể, đảm bảo mọi thành phần được xây dựng trên đám mây đều có một nền tảng an toàn, nhất quán, dễ kiểm toán và hiệu quả ngay từ đầu.

Các thành phần và chức năng chính của kiến trúc:

- Quản lý định danh và truy cập: Sử dụng một hệ thống quản lý tập trung để cấp quyền cho người dùng, áp dụng xác thực đa yếu tố và nguyên tắc đặc quyền tối thiểu.

- An ninh mạng: Phân chia mạng thành các vùng riêng biệt hoặc cho phép truy cập từ xa. Toàn bộ lưu lượng ra vào đều được kiểm soát chặt chẽ qua tường lửa và hệ thống phát hiện/phòng chống xâm nhập (IDS/IPS).

- Bảo vệ dữ liệu: Mã hóa mọi dữ liệu cả khi lưu trữ và khi đang truyền. Hệ thống quản lý khóa mã hóa được bảo vệ nghiêm ngặt.

- Giám sát và ghi nhật ký: Tất cả các hoạt động, các thay đổi cấu hình, các lượt truy cập đều được ghi lại và đưa về một nơi lưu trữ tập trung, an toàn để phục vụ việc điều tra và phân tích an ninh.

- Quản trị và tuân thủ: Thiết lập các quy tắc để tự động ngăn chặn các hành động cấu hình không được truy cập.

Vấn đề an toàn và kiến trúc để đảm bảo an toàn mạng cho hệ thống thông tin

- Phòng thủ nhiều lớp (Defense-in-Depth): Kiến trúc an ninh mạng cho phép thiết kế một hệ thống phòng thủ nhiều lớp, từ bảo mật vật lý, mạng, ứng dụng đến dữ liệu. Chuẩn an ninh như ISO/IEC 27002 hướng dẫn về việc triển khai các biện pháp bảo mật tại mỗi lớp [59, 60].

- Bảo mật trong hệ thống đám mây: Kiến trúc bảo mật trên đám mây như mô hình Shared Responsibility Model của AWS, kết hợp với các chuẩn như ISO/IEC 27017, đảm bảo rằng dữ liệu trên đám mây được bảo vệ trong mọi tình huống [20].

- Xử lý và ứng phó sự cố: Chuẩn NIST SP 800-61 (Computer Security Incident Handling Guide) giúp tổ chức xây dựng một quy trình chuẩn để phát hiện, ứng phó và phục hồi sau sự cố bảo mật, giảm thiểu tác động và rủi ro [76].

Thách thức trong việc áp dụng kiến trúc và chuẩn an ninh mạng bao gồm:

- Độ phức tạp của hệ thống: Các hệ thống lớn, phân tán làm tăng độ khó trong việc triển khai kiến trúc bảo mật toàn diện.

- Chi phí: Việc áp dụng các chuẩn an ninh mạng đôi khi đòi hỏi đầu tư lớn vào công nghệ và nhân sự.

- Cập nhật liên tục: Các chuẩn an ninh và kiến trúc bảo mật phải liên tục được cập nhật để đối phó với các mối đe dọa mới.

1.1.2 Vấn đề quản trị mạng hiệu quả cho mạng cục bộ và đám mây

Vấn đề quản trị các mạng với giao thức SNMP và CMIP

Đối với mạng máy tính gồm các yêu cầu: phân tích và thiết lập mạng; phát triển các ứng dụng trên mạng; quản trị mạng. Trong đó, nhu cầu quản trị hệ thống mạng máy tính rất quan trọng do:

- Sự tăng không ngừng về nhu cầu xử lý thông tin của các tổ chức đã đòi hỏi phát triển nhanh chóng công nghệ máy tính và mạng dữ liệu, nhằm hỗ trợ các nhu cầu đa dạng về xử lý thông tin [8].

- Các loại thiết bị mạng và chủng loại các mạng máy tính ngày càng đa dạng, không chỉ theo hãng sản xuất mà còn sử dụng các kiểu trúc khác nhau.

- Dưới con mắt của nhà quản trị hệ thống, hạ tầng phần mềm cũng cần được thiết kế và thể hiện theo nhiều loại đa dạng và khác nhau [8].

- Việc quản trị mạng máy tính được thực hiện theo những chuẩn mực, tạo điều kiện thuận lợi cho việc xác định cấu hình thiết bị, giá thiết bị, lựa chọn chức năng của các thiết bị…

Trong nhiều năm qua, quản trị hệ thống mạng máy tính sử dụng 2 bộ giao thức quản lý mạng sau:

Giao thức SNMP (Simple Network Management Protocol): SNMP là giao thức quản lý mạng phổ biến, được thiết kế để quản lý và giám sát các thiết bị mạng như bộ định tuyến, bộ chuyển mạch, và server. SNMP sử dụng mô hình quản lý tập trung, với các thành phần chính gồm: Manager (Trình quản lý): Điều khiển và thu thập thông tin từ các thiết bị mạng; Agent (Tác tử): Chạy trên các thiết bị được quản lý để thu thập dữ liệu và phản hồi yêu cầu từ Manager; MIB (Management Information Base): Cơ sở dữ liệu chứa các thông tin về các đối tượng mạng có thể được quản lý [92].

Giao thức CMIP (Common Management Information Protocol): CMIP là một giao thức quản lý mạng tiên tiến hơn, được thiết kế như một sự thay thế cho SNMP. CMIP hoạt động dựa trên mô hình OSI và cung cấp các tính năng quản lý mạng mạnh

mẽ hơn, bao gồm: Hỗ trợ đối tượng hướng đối tượng: CMIP mô tả các thực thể mạng dưới dạng các đối tượng với các thuộc tính và hành vi; Khả năng kiểm soát mạnh mẽ: CMIP hỗ trợ nhiều thao tác hơn, như hành động (action) và ràng buộc (constraint), giúp tăng cường khả năng quản lý [90].

Việc lựa chọn giao thức SNMP và CMIP làm cặp giao thức chính để nghiên cứu tích hợp xuất phát từ những lý do như sau:

- Giao thức SNMP: Là giao thức theo mô hình TCP/IP, trở thành một chuẩn de facto (chuẩn thực tế) nhờ vào sự đơn giản, gọn nhẹ, dễ triển khai và được hỗ trợ bởi hầu hết các thiết bị mạng. Nó cực kỳ mạnh mẽ trong việc giám sát và thu thập trạng thái.

- Giao thức CMIP: Là giao thức theo mô hình OSI, một chuẩn được thiết kế với cấu trúc hướng đối tượng chặt chẽ, cung cấp năng lực điều khiển và thực thi các tác vụ quản trị phức tạp vượt trội so với SNMP.

Sự đối lập giữa "đơn giản, phổ biến" và "mạnh mẽ, phức tạp" này tạo ra một bài toán nghiên cứu kinh điển và hấp dẫn: Làm thế nào để kết hợp thế mạnh của cả hai? Việc tích hợp chúng không chỉ là một thách thức kỹ thuật mà còn là một vấn đề khoa học, nhằm tạo ra một giải pháp quản trị toàn diện hơn. Khả năng bổ sung cho nhau: Chính vì sự khác biệt, hai giao thức này có khả năng bổ sung hoàn hảo cho nhau. Một hệ thống tích hợp có thể tận dụng khả năng giám sát rộng khắp, chi phí thấp của SNMP trên toàn mạng, đồng thời sử dụng khả năng điều khiển mạnh mẽ, an toàn của CMIP cho các tác vụ quản trị quan trọng hoặc trên các thiết bị lõi đòi hỏi sự can thiệp phức tạp.

Trong khi đó, các giao thức mạng và quản lý mạng mới như:

- ***Giao thức SSH (Secure Shell)***: chủ yếu được sử dụng để truy cập và cấu hình thiết bị một cách thủ công qua giao diện dòng lệnh (CLI). Nó là công cụ để quản trị trực tiếp, có tương tác, không phải là một giao thức cho hệ thống quản trị mạng (NMS) tự động thu thập dữ liệu hiệu năng trên quy mô lớn.

- ***Giao thức sFlow/NetFlow***: là các giao thức chuyên dụng cho việc phân tích luồng lượng (traffic flow analysis). Chúng trả lời câu hỏi "ai đang nói chuyện với ai

và nói gì?" chứ không tập trung vào việc giám sát trạng thái sức khỏe của bản thân thiết bị (CPU, RAM, nhiệt độ, trạng thái cổng...).

- ***Giao thức HTTP/HTTPS***: thường được dùng để cung cấp giao diện quản trị dựa trên web (Web-based GUI) cho người dùng cuối. Nghiên cứu về SNMP và CMIP giúp giải quyết bài toán tích hợp giữa hai framework quản trị mạng có cấu trúc và mô hình thông tin (Information Model) hoàn toàn khác biệt, một bài toán mang tính nền tảng hơn.

Sự phát triển của của mạng đám mây và SDN đã thay đổi mạnh mẽ cách thiết kế và vận hành mạng: tài nguyên hạ tầng được ảo hoá, khả năng lập trình, và các nền tảng điều khiển quản trị dịch chuyển theo hướng tự động hoá, cloud native và mô hình hoá đã thay đổi cách tiếp cận cách quản trị mạng trong nghiên cứu các nghiên cứu mới:

- ***NETCONF và YANG***: trở thành tiêu chuẩn de facto cho cấu hình: Các bài báo gần đây đặc biệt nhấn mạnh vai trò của NETCONF/RESTCONF cùng với ngôn ngữ mô hình dữ liệu YANG. Một bài báo trên Tạp chí JOCN năm 2021 đã trình bày chi tiết về YANG và các giao thức liên quan như NETCONF/RESTCONF như là nền tảng cho việc quản lý mạng quang thế hệ mới. YANG cho phép mô tả cấu hình và trạng thái một cách có cấu trúc, khắc phục nhược điểm của MIB trong SNMP.

- ***Telemetry Streaming***: là xu hướng nghiên cứu quản trị mạng trên SDN hứa hẹn đem lại hiệu năng cao do:

Khắc phục nhược điểm của SNMP Polling: Các bài báo khoa học chỉ ra rõ ràng rằng SNMP hoạt động theo mô hình "pull" (kéo dữ liệu), trong khi Telemetry sử dụng mô hình "push" (đẩy dữ liệu). Mô hình push giúp giảm tải cho mạng và cho phép thu thập dữ liệu với tần suất cao hơn nhiều (thậm chí microsecond) mà không làm quá tải thiết bị .

Hiệu suất vượt trội được chứng minh bằng thực nghiệm: Nghiên cứu thực nghiệm trên mạng quang SDN năm 2021 đã so sánh hiệu năng của các giao thức telemetry (gRPC, gNMI, YANG Push) với các phương pháp truyền thống. Kết quả cho thấy telemetry có độ trễ (latency) và chi phí overhead thấp hơn đáng kể, đáp ứng được yêu cầu khắt khe của các mạng thế hệ mới.

Tích hợp với lập trình được (Programmability): Telemetry gắn liền với khái niệm "dữ liệu phẳng có thể lập trình được" (Programmable Data Planes) sử dụng ngôn ngữ P4. Các bài báo mô tả việc nhúng thông tin telemetry trực tiếp vào gói tin (In-band Network Telemetry - INT) cho phép giám sát từng bước đường đi của gói tin với độ chính xác cao.

Tuy nhiên, trong đa số hệ thống thực tế (đặc biệt ở doanh nghiệp và mạng viễn thông), quá trình dịch chuyển thường được diễn ra theo hướng môi trường lai gồm:

- Hạ tầng vật lý bên dưới có vòng đời dài như bộ định tuyến (Router), Bộ chuyển mạch (Switch), tường lửa (Firewall),…
- Các miền mạng/thiết bị mạng cũ đang vận hành ổn định;
- Các miền cloud-native/SDN mới được đưa vào từng phần;
- Các công cụ quản trị mạng cũ và mới tồn tại theo nhiều thế hệ.

Trong bối cảnh đó, bài toán quản trị mạng không chỉ là "chọn giao thức quản trị mới", mà là đảm bảo khả năng quản trị trên một hệ thống đa miền và đa thế hệ, trong đó tích hợp trở thành yêu cầu bắt buộc: tích hợp dữ liệu, tích hợp mô hình quản lý, tích hợp cơ chế cảnh báo giám sát, và tích hợp chính sách vận hành bảo mật.

Việc nghiên cứu theo hướng tích hợp SNMP/CMIP nhằm giải quyết một nhu cầu thực tế: làm cầu nối giữa thiết bị mạng cũ và các thiết bị mạng mới giúp vận hành liên tục, giảm chi phí chuyển đổi, đồng thời tạo lộ trình nâng cấp từng phần.

Vấn đề quản trị mạng bằng công nghệ tác tử

Tác tử di động (Mobile Agent) là các chương trình phần mềm tự chủ, có khả năng di chuyển giữa các nút mạng, thực hiện các tác vụ độc lập, và giao tiếp với các thành phần khác để hoàn thành nhiệm vụ. Khả năng tự di chuyển và hoạt động độc lập của tác tử di động đã mở ra nhiều hướng tiếp cận mới trong quản trị mạng, đặc biệt là đối với các mạng phân tán lớn và phức tạp [66].

Lợi ích của áp dụng tác tử di động vào trong quản trị mạng:

Giảm tải lưu lượng mạng: Thay vì gửi toàn bộ dữ liệu từ các nút mạng về trung tâm để xử lý, Tác tử di động chỉ cần truyền tải kết quả của các tác vụ đã xử lý cục bộ, qua đó giảm đáng kể lưu lượng mạng và tăng hiệu quả truyền tải [64].

Tính linh hoạt và khả năng mở rộng: Tác tử di động có thể được lập trình để thực hiện nhiều nhiệm vụ khác nhau và dễ dàng mở rộng quy mô theo nhu cầu của hệ thống mạng. Điều này đặc biệt quan trọng trong các mạng phân tán, không đồng nhất [66].

Hoạt động trong môi trường không liên tục: Tác tử di động có thể thực hiện các nhiệm vụ trong môi trường không có kết nối mạng liên tục. Khi di chuyển đến nút đích, chúng thực hiện nhiệm vụ cục bộ mà không phụ thuộc vào sự giám sát từ máy chủ trung tâm [81].

Phát hiện và xử lý sự cố nhanh chóng: Tác tử di động có thể được triển khai để phát hiện các sự cố mạng và thực hiện các biện pháp khắc phục trực tiếp tại nút bị ảnh hưởng. Điều này giúp giảm thời gian phản ứng và nâng cao độ tin cậy của hệ thống [64].

Vấn đề đảm bảo an toàn cho môi trường tác tử di động

Ứng dụng tác tử di động trong quản lý hệ thống mạng giúp hệ thống mạng an toàn và bảo mật hơn. Tuy nhiên, bản thân nền tảng tác tử di động cũng ẩn chứa nguy cơ mất an toàn và an ninh thông tin. Một số vấn để cần đảm bảo an toàn thông tin cho môi trường tác tử di động phát sinh và cần đảm bảo khi ứng dụng như:

Đánh giá rủi ro và phân tích mối nguy: Để đảm bảo an toàn cho môi trường tác tử di động, bước đầu tiên là đánh giá rủi ro và phân tích các mối nguy tiềm ẩn. Các tác tử di động thường hoạt động trong môi trường mạng không dây, nơi dễ bị tấn công bởi các mối đe dọa như nghe lén, giả mạo, và từ chối dịch vụ (DoS). Việc xác định các điểm yếu trong hệ thống và các kịch bản tấn công có thể xảy ra là cần thiết để thiết kế các biện pháp phòng ngừa hiệu quả [93].

Mã hóa dữ liệu và bảo mật thông tin: Mã hóa dữ liệu là một trong những phương pháp cơ bản để bảo vệ thông tin truyền giữa các tác tử di động và hệ thống. Các thuật toán mã hóa như AES (Advanced Encryption Standard) và RSA (Rivest-Shamir-Adleman) được sử dụng rộng rãi để đảm bảo tính bí mật và toàn vẹn của dữ liệu. Ngoài ra, việc áp dụng các giao thức bảo mật như TLS (Transport Layer Security) cũng giúp ngăn chặn các cuộc tấn công nghe lén và giả mạo [71].

Xác thực và quản lý danh tính: Xác thực là quá trình quan trọng để đảm bảo chỉ các tác tử di động hợp pháp mới có thể truy cập vào hệ thống. Các phương pháp xác thực như mật khẩu, chứng thư, và sinh trắc học có thể được sử dụng. Đồng thời, việc quản lý danh tính thông qua các hệ thống như PKI (Public Key Infrastructure) giúp kiểm soát quyền truy cập và giảm thiểu rủi ro từ các tác tử độc hại [80].

Giám sát và phát hiện xâm nhập: Hệ thống giám sát và phát hiện xâm nhập (IDS) là công cụ quan trọng để phát hiện các hoạt động bất thường trong môi trường tác tử di động. IDS có thể dựa trên signature (chữ ký) hoặc anomaly (bất thường) để nhận diện các cuộc tấn công tiềm ẩn. Việc tích hợp IDS với các hệ thống phản ứng tự động (IPS - Intrusion Prevention System) giúp ngăn chặn kịp thời các mối đe dọa [86].

Cập nhật và vá lỗi bảo mật: Các lỗ hổng bảo mật trong phần mềm tác tử di động và hệ thống mạng có thể bị khai thác bởi tin tặc. Do đó, việc thường xuyên cập nhật phần mềm và vá lỗi là cần thiết để giảm thiểu rủi ro. Các nhà phát triển cần theo dõi các bản cập nhật bảo mật từ nhà cung cấp và triển khai chúng kịp thời [47].

Quản lý quyền truy cập và phân quyền: Việc quản lý quyền truy cập dựa trên vai trò (RBAC - Role-Based Access Control) giúp hạn chế quyền truy cập của các tác tử di động chỉ trong phạm vi cần thiết. Điều này giảm thiểu thiệt hại trong trường hợp một tác tử bị xâm nhập. Các cơ chế phân quyền cần được thiết kế chặt chẽ để đảm bảo tính an toàn của hệ thống [83].

Sử dụng mạng riêng ảo (VPN): VPN là công cụ hiệu quả để bảo vệ kết nối giữa các tác tử di động và hệ thống trung tâm. Bằng cách tạo ra một đường truyền mã hóa, VPN giúp ngăn chặn việc đánh cắp thông tin và đảm bảo tính riêng tư của dữ liệu. Các giao thức VPN như IPsec và OpenVPN được khuyến nghị sử dụng [38].

Đào tạo và nâng cao nhận thức: Nhận thức về bảo mật của người dùng và nhà phát triển đóng vai trò quan trọng trong việc đảm bảo an toàn cho môi trường tác tử di động. Các chương trình đào tạo về bảo mật thông tin và thực hành an toàn mạng cần được triển khai thường xuyên để giảm thiểu các lỗi do con người gây ra [79].

1.2 Tổng quan các nghiên cứu về quản trị mạng cục bộ và đám mây

Nghiên cứu sử dụng giao thức SNMP

Các nghiên cứu của Alhilali, Ahmed và cộng sự [15] về các vấn đề mà quản trị viên mạng thường gặp phải bao gồm hiệu suất mạng không hiệu quả, chẳng hạn như việc sử dụng băng thông không tối ưu và việc giám sát tài nguyên mạng phức tạp. Để duy trì sự ổn định của hạ tầng mạng Internet, việc giám sát thời gian thực là cần thiết. Nghiên cứu này nhằm cung cấp một thiết kế khái niệm cho hệ thống Chất lượng dịch vụ (QoS) và Giao thức quản lý mạng đơn giản (SNMP) để quản lý băng thông Internet và giám sát tài nguyên mạng. Kết quả cho thấy việc sử dụng băng thông tại các tổ chức, doanh nghiệp đã vượt quá giới hạn, nhưng có thể phục hồi nhờ vào hệ thống QoS.

Đồng thời, trong nghiên cứu về băng thông mạng và các công cụ quản lý mạng của Breaban [25], Diana [30], Barreiros [23], Simsek [89] cũng đã đề cập về sử dụng giao thức SNMP để quản lý hệ thống mạng đơn giản và hiệu quả, tuy nhiên chưa đề xuất phương thức quản lý mạng mới.

Ya-shiang peng, Yen-cheng Chen [103], J. Swarna, C. Senthil raja, Dr.K.S.ravi chandran, Laurent Andrey, Olivier Festoro [64] nghiên cứu về ứng dụng giao thức quản lý SNMP trong Mạng đám mây nhưng cũng chưa đề xuất về tích hợp giao thức CMIP.

Trong các hệ thống quản lý và giám sát mạng, hay Trạm quản lý mạng (NMS), Giao thức giám sát mạng đơn giản (SNMP) thường được sử dụng, với giao thức này, người ta có thể thu thập thông tin về hành vi, giá trị của các biến và trạng thái của kiến trúc mạng. Tuy nhiên, đối với các mạng doanh nghiệp lớn, giao thức này có thể gây ra độ trễ trong việc thu thập và xử lý dữ liệu, do đó làm cho việc giám sát theo thời gian thực trở nên khó khăn. Bài báo [36] của Espinel Villalobos, R.I., Ardila Triana, E., Zarate Ceballos, đề xuất một hệ thống đa tác tử dựa trên các lớp, với ba loại tác tử. Điều này bao gồm tác tử thu thập, sử dụng giá trị Cơ sở thông tin quản lý (MIB) để thu thập thông tin từ thiết bị mạng, bảng thông tin đầu vào từ các thiết bị mạng để tác tử hợp nhất xử lý dữ liệu đã thu thập và để lại ở định dạng có thể sử dụng được, và sau đó được tác tử ứng dụng biểu diễn dưới dạng dịch vụ web, trong trường

hợp này là bản đồ nhiệt. Nghiên cứu này có hướng nghiên cứu mới, nhưng chỉ đang áp dụng cho giao thức SNMP, chưa có hướng nghiên cứu cho giao thức CMIP.

Nghiên cứu của Alhilali, A.H. [14] tập trung vào việc thiết kế và triển khai hệ thống quản lý lưu lượng máy chủ đến bằng Giao thức quản lý mạng đơn giản (SNMP). Bản chất của phương pháp này là cung cấp hệ thống giám sát thời gian thực giúp người quản lý mạng giám sát hiệu quả mọi hoạt động mạng. Điểm khác biệt chính của hệ thống được phát triển nằm ở việc sử dụng giao thức SNMP để giám sát và kiểm soát mạng mà không cần đến công cụ của bên thứ ba. Phương pháp đề xuất giải quyết các thách thức chung về mạng của người quản trị, bao gồm việc sử dụng băng thông không hiệu quả và giám sát tài nguyên mạng phức tạp. Sau khi triển khai, hệ thống đã thu thập thông tin thiết bị mạng cần thiết từ Cơ sở thông tin quản lý (MIB). Ngoài ra, khả năng cảnh báo của hệ thống giúp tăng cường khả năng phục hồi của mạng trước các bất thường về lưu lượng đến có thể làm gián đoạn hoạt động của máy chủ. Nghiên cứu này đang áp dụng cho giao thức SNMP, chưa có hướng nghiên cứu cho giao thức CMIP.

Nghiên cứu mô hình và tối ưu quản trị mạng mới

Một số hướng quản lý mạng đề xuất mới theo mô hình lý thuyết như các bài nghiên cứu như M. and Lundqvist [23] cũng đề cập tới việc xây dựng một bản thể mạng (ontology) một hệ thống quản lý mạng được đơn giản hóa; theo hướng sử dụng các phương pháp tiếp cận dựa trên OntoUML hoặc UFO. Và đồng thời đề xuất một phương pháp tiếp cận tinh chỉnh cho các ontology được chỉ định trong OntoUML. Phương pháp tiếp cận tinh chỉnh này cho một số cạm bẫy của việc kế thừa như được sử dụng trong lập trình và đặc tả hướng đối tượng "cổ điển" nói chung và trong OntoUML nói riêng nhưng chưa có hiện thực hoá cài đặt cụ thể.

Đề xuất nén lưu lượng sử dụng thuật toán mã hoá sử dụng Mạng nơ-ron đồ thị không gian-thời gian (ST-GNN), để mô hình hóa các mẫu này và đạt được tỷ lệ nén vượt trội so với các phương pháp truyền thống như GZIP [77]. Bài nghiên cứu của Paul Almasan và các đồng nghiệp tập trung vào hai kịch bản nén: nén liên kết đơn, khai thác tương quan thời gian và nén toàn mạng, khai thác cả tương quan không gian và thời gian. Các tác giả chứng minh rằng phương pháp dựa trên ST-GNN của họ

vượt trội đáng kể so với GZIP và thậm chí cả các phương pháp dựa trên RNN, làm nổi bật hiệu quả của việc khai thác cả mẫu không gian và thời gian để nén lưu lượng. Nghiên cứu này xây dựng hướng tối ưu quản lý mạng thông qua phương pháp nén dữ liệu truyền, đây là một hướng giúp tiết kiệm băng thông mạng.

Simsek, G., Ergenç, D., & Onur đề xuất giải pháp giám sát mạng truyền thống thường thiếu khả năng mở rộng do bản chất tập trung của chúng là thu thập nhịp tim từ tất cả các thành phần mạng thông qua một bộ điều khiển duy nhất [89]. Là một giải pháp, khuôn khổ In-Band Network Telemetry (INT) gần đây đã được đề xuất để thu thập thông tin đo từ xa mạng một cách tự chủ hơn và phân tán hơn bằng cách sử dụng các công tắc có thể lập trình. Tuy nhiên, nó đặt ra thêm những thách thức để (i) tìm các đường dẫn INT phù hợp để tối ưu hóa chi phí kiểm soát và độ mới của thông tin và (ii) đảm bảo phân phối thông tin kiểm soát đáng tin cậy qua các đường dẫn INT đa bước.

Một số bài báo nghiên cứu về tích hợp các giao thức quản lý mạng đơn giản SNMP và giao thức CMIP cho phép hệ thống tích hợp được đa dạng các thiết bị trong thực tế và đáp ứng các chuẩn ISO/CCITT về quản lý mạng.

Các tiêu chuẩn mạng LAN/MAN của tổ chức IEEE về MIB [57], tiêu chuẩn về giao thức CMIP CCITT, X.710 and ISO/IEC 9595 [26] và một số về nghiên cứu tích hợp quản lý mạng khác.

Bài báo [81] của Saydam, T., & Sirsikar, R. và Zihang, R., & Lobelle, Wang [100], M [107] và Koerner cũng có mô tả các thông số kỹ thuật và thiết kế cần thiết của một cổng ứng dụng CMIP-SNMPv2 để tích hợp và khả năng tương tác quản lý mạng. Một thiết bị phần mềm proxy trung gian như vậy cho phép quản lý các đối tượng Internet MIB-II thông qua trình quản lý ISO/CCITT hỗ trợ giao thức CMIP và các dịch vụ quản lý CMIS. Tất cả các chức năng của cổng được giải thích bằng sơ đồ luồng dữ liệu. Các PDU chỉ báo CMIP được chuyển đổi thành các PDU SNMPv2 tương ứng về mặt chức năng. Các khía cạnh ánh xạ dịch vụ và ánh xạ tên chi tiết phù hợp với các giao thức, Internet MIB-II và GDMO MIB được khám phá đầy đủ. Ánh xạ ngược SNMPv2 sang CMIP cũng được giải thích chi tiết, tiếp theo là các kết luận nắm bắt các khía cạnh nổi bật của thiết kế này nhưng chưa có đề xuất kết hợp giải pháp ứng dụng Tác tử di động vào tích hợp.

Nghiên cứu về giám sát hệ thống

Một số nghiên cứu hệ thống giám sát giao thông minh trong việc sử dụng PLC và SCADA [73], giải pháp giao thông thông minh của VNSMAPS [5] sử dụng thiết bị của Soway Tech [91] và nghiên cứu IoT và BigData trong xử lý hệ thống giao thông thông minh đưa ra giải pháp ứng dụng vào hệ thống giao thông thông minh nhưng chưa tối ưu về băng thông và tính bảo mật của hệ thống, sử dụng hệ thống GPS cho các phương tiện khẩn cấp [11], dữ liệu về trọng tải của phương tiện [72]. Nghiên cứu áp dụng nền tảng MapReduce vào xử lý dữ liệu [62] để tối ưu hệ thống, thiết kế hạ tầng khoá công khai di động M-PKI [84] giúp cho hệ thống bảo mật hơn nhưng cho đưa ra giải pháp tổng thể chung.

Nghiên cứu về mạng SDN

Các bài báo nghiên cứu tổng quan về SDN [94, 101] phân tích đe dọa bảo mật đối với các mặt phẳng ứng dụng, điều khiển và dữ liệu của SDN. Các nền tảng bảo mật bảo vệ từng mặt phẳng được mô tả theo sau là các phương pháp bảo mật khác nhau để bảo mật toàn mạng trong SDN. Bảo mật SDN được phân tích theo các chiều bảo mật của khuyến nghị ITU-T, cũng như theo chi phí của các giải pháp bảo mật. Tóm lại, bài báo này nêu bật những thách thức bảo mật hiện tại và tương lai trong SDN và các hướng đi trong tương lai cho SDN an toàn.

1.3 Tổng quan ứng dụng tác tử di động vào quản trị mạng cục bộ và đám mây

Các đặc trưng cốt lõi của môi trường đám mây [70]

- Tự phục vụ theo yêu cầu : Người dùng có thể đơn phương cung cấp các tài nguyên tính toán (máy chủ, lưu trữ, mạng) một cách tự động mà không cần đến sự tương tác của con người với nhà cung cấp dịch vụ. Đây là sự khác biệt cơ bản với mô hình IT truyền thống, nơi việc cấp phát một máy chủ mới có thể mất hàng ngày hoặc hàng tuần.

- Truy cập mạng rộng khắp: Các tài nguyên có sẵn trên mạng và được truy cập thông qua các cơ chế tiêu chuẩn, cho phép sử dụng bởi các nền tảng khách hàng không đồng nhất (ví dụ: điện thoại di động, máy tính xách tay, máy trạm).

- Gom cụm tài nguyên: Các tài nguyên tính toán của nhà cung cấp được gom lại để phục vụ nhiều người tiêu dùng theo mô hình đa người thuê. Tài nguyên vật lý và

ảo được gán và tái gán một cách linh hoạt theo nhu cầu. Người dùng thường không có quyền kiểm soát hoặc biết về vị trí chính xác của các tài nguyên được cung cấp.

- Co giãn linh hoạt nhanh chóng: Tài nguyên có thể được cung cấp và giải phóng một cách linh hoạt, trong một số trường hợp là tự động, để nhanh chóng tăng hoặc giảm quy mô theo nhu cầu. Đối với người tiêu dùng, các tài nguyên có vẻ như là vô hạn và có thể được mua với số lượng bất kỳ, bất kỳ lúc nào.

- Dịch vụ được đo lường: Hệ thống đám mây tự động kiểm soát và tối ưu hóa việc sử dụng tài nguyên bằng cách tận dụng khả năng đo lường ở một mức độ trừu tượng phù hợp với loại dịch vụ (ví dụ: lưu trữ, xử lý, băng thông). Việc sử dụng tài nguyên có thể được theo dõi, kiểm soát và báo cáo, mang lại sự minh bạch cho cả nhà cung cấp và người tiêu dùng.

- Mô hình trách nhiệm chia sẻ: Đây là đặc trưng then chốt về mặt an ninh và vận hành. Nhà cung cấp đám mây chịu trách nhiệm bảo mật đám mây (hạ tầng vật lý, ảo hóa), trong khi khách hàng chịu trách nhiệm bảo mật về dữ liệu, ứng dụng, cấu hình mạng và tường lửa. Việc không hiểu rõ mô hình này là nguyên nhân hàng đầu dẫn đến các sự cố bảo mật.

Các vấn đề kỹ thuật cụ thể khi dịch chuyển mạng cục bộ lên đám mây

Dựa trên các đặc trưng trên, việc di trú một mạng cục bộ không đơn thuần là đưa lên đám mây mà phải đối mặt và giải quyết các vấn đề kỹ thuật sau:

- Thiết kế lại hoàn toàn Kiến trúc mạng: Một cấu hình sai trong bộ định tuyến, Bộ chuyển mạch có thể vô tình phơi bày toàn bộ tài nguyên nội bộ ra Internet, một rủi ro khó xảy ra hơn trong mạng cục bộ vật lý được kiểm soát chặt chẽ.

- Sự thay đổi căn bản trong Mô hình an ninh: Việc bảo vệ một máy chủ không còn chỉ là đặt nó sau một tường lửa vật lý, mà là định nghĩa chính xác vai trò nào được phép truy cập, từ những nguồn nào và được thực hiện những hành động gì.

- Kết nối giữa mạng cục bộ và đám mây: Lựa chọn sai phương án có thể dẫn đến thắt cổ chai băng thông, độ trễ cao làm ảnh hưởng đến trải nghiệm người dùng, hoặc các rủi ro bảo mật nếu kết nối không được mã hóa đúng cách.

- Hiệu năng và độ trễ ứng dụng: Nhiều ứng dụng cũ được thiết kế để hoạt động trong môi trường cục bộ có độ trễ cực thấp. Khi một phần ứng dụng được đưa lên

đám mây và phần còn lại vẫn ở cục bộ, độ trễ hàng chục ms qua kết nối mạng có thể làm cho các ứng dụng hoạt động cực kỳ chậm chạp hoặc thậm chí không thể sử dụng. Đây là một vấn đề kỹ thuật thuần túy, đòi hỏi phải phân tích kỹ lưỡng luồng dữ liệu của ứng dụng trước khi quyết định di trú thành phần nào.

Nghiên cứu ứng dụng Tác tử di động

Ichiro Satoh có một số bài nghiên cứu về Tác tử di động [54], và ứng dụng của Tác tử di động [49-54] cũng như ứng dụng Tác tử di động trong quản lý mạng [55, 56], các nghiên cứu của Ichiro Satoh giới thiệu các thuộc tính và phạm vi ứng dụng của tác tử di động. Các nghiên cứu về nền tảng tác tử di động và tài liệu lập trình nền tảng tác tử di động nguồn mở JADE [38], tiêu chuẩn FIPA [41] cung cấp tài liệu lập trình nền tảng tác tử di động và chuẩn quốc tế cho nền tảng tử di động .

Bài báo của A.K. Sharma1, Atul Mishra, Vijay Singh [9] này đề xuất một kiến trúc quản lý mạng mới có tên là CNMA, được thiết kế để giải quyết các hạn chế về khả năng mở rộng và tính linh hoạt của các hệ thống quản lý mạng tập trung truyền thống. Các tác giả lập luận rằng tính phức tạp và quy mô ngày càng tăng của các mạng hiện đại đòi hỏi một phương pháp tiếp cận phân tán, đặc biệt là phương pháp tận dụng sức mạnh của các tác tử di động. CNMA chia mạng thành các mạng con do các tác tử di động quản lý được gọi là M-SNLM. Các tác tử này có thể tạo ra các M-SNLM con một cách động khi mạng phát triển, tạo ra một cấu trúc quản lý phân cấp. Bài báo nêu bật hiệu quả của CNMA bằng cách so sánh chi phí quản lý mạng của nó với các mô hình tác tử tập trung và phẳng truyền thống. Thông qua phân tích chi phí sử dụng mạng mô phỏng, các tác giả chứng minh rằng CNMA làm giảm đáng kể dữ liệu được truyền để quản lý mạng, khiến nó trở thành một giải pháp hấp dẫn để quản lý các mạng lớn, phức tạp và năng động tuy nhiên chưa đầy đủ và khả năng tích hợp các giao thức quản lý mạng SNMP và CMIP cũng như tối ưu hệ thống và đảm bảo an toàn bảo mật cho hệ thống.

Bài báo [22] Anish Saini và Atul Mishra đề xuất một cách tiếp cận mới đối với quản lý mạng được gọi là "mô hình quản lý mạng phân vùng miền" dựa trên các tác tử di động và hệ thống quản lý phần tử (EMS). Các tác giả lập luận rằng các mô hình

quản lý mạng tập trung hiện có, chẳng hạn như các mô hình dựa trên SNMP (Giao thức quản lý mạng đơn giản), thiếu khả năng mở rộng và linh hoạt. Họ đề xuất một giải pháp sử dụng các tác tử di động, là các chương trình có thể di chuyển tự động giữa các thiết bị mạng. Hệ thống được đề xuất chỉ định một tác tử thông minh cho mỗi EMS, cho phép quản lý cục bộ và báo cáo kết quả cho một người quản lý toàn cầu. Cấu trúc phân cấp này giảm thiểu luồng dữ liệu quản lý đến một máy chủ tập trung và tăng cường hiệu quả, khả năng mở rộng và tính linh hoạt. Các tác giả so sánh mô hình của họ với các mô hình quản lý mạng dựa trên tác tử di động khác, nêu bật các ưu điểm của nó và trình bày các kết quả thử nghiệm để hỗ trợ cho các tuyên bố của họ.

Bài báo [61] nghiên cứu về Dịch vụ và Kiến trúc khám phá những hạn chế của Giao thức Quản lý Mạng Đơn giản (SNMP) trong bối cảnh lưu lượng mạng đám mây tăng nhanh. Các tác giả xác định những điểm yếu chính của SNMP, bao gồm sự phụ thuộc vào truyền tải UDP không đáng tin cậy, việc thăm dò định kỳ có thể dẫn đến dữ liệu không chính xác và lưu lượng mạng cao, và việc thiếu các tính năng bảo mật mạnh mẽ. Để giải quyết những thiếu sót này, bài báo đề xuất một Mô hình Quản lý Mạng đám mây (CNMM) mới. Mô hình này nhằm mục đích cải thiện hiệu quả truyền thông và bảo mật bằng cách giảm thiểu trao đổi gói tin giữa người quản lý và tác tử, cung cấp các bản cập nhật định kỳ mà không dựa vào các truy vấn của người quản lý và triển khai kiến trúc máy chủ quản lý ảo hóa. Mô hình được đề xuất tận dụng khái niệm cập nhật dựa trên ngưỡng và thông báo bẫy, nâng cao hiệu suất, bảo mật và khả năng quản lý tổng thể. Mặc dù mô hình hiện là một khái niệm lý thuyết, các tác giả nhấn mạnh tiềm năng phát triển trong tương lai của nó, với các kế hoạch khám phá thiết kế và triển khai chi tiết, cũng như các ứng dụng tiềm năng của công nghệ OpenFlow.

Zubair, M., & Manzoor [107] nghiên cứu về tác tử di động, có nhận định một trong những tiến bộ trong công nghệ tác tử di động trong những thập kỷ gần đây là việc sử dụng chúng trong các ứng dụng quản lý mạng vì bản chất phân tán và có thể mở rộng của chúng. Tuy nhiên, một trong những mối quan tâm chính trong việc triển

khai thực tế các phương pháp tiếp cận này là thiếu cơ chế chịu lỗi. Có một số kỹ thuật chịu lỗi tổng quát có thể được điều chỉnh trong các mô hình quản lý mạng dựa trên tác tử di động để đáp ứng các yêu cầu chịu lỗi. Trong bài báo này trình bày một cuộc khảo sát về các kỹ thuật chịu lỗi hiện có và các mô hình kiến trúc quản lý mạng dựa trên tác tử di động khác nhau, trong đó các kỹ thuật này có thể được áp dụng.

Để nâng cao hiệu quả quản lý mạng truyền thống, công nghệ Tác tử di động được áp dụng trong quản lý mạng. Tian-jun [96] đề xuất theo ý tưởng mã hóa riêng biệt, mô hình quản lý mạng Tác tử di động lồng nhau được xây dựng bởi Agent tác vụ và Tác tử điều hướng. Công thức tính thời gian sử dụng của Tác tử di động được thiết lập và một số thí nghiệm được tiến hành so sánh bằng cách sử dụng phong cách quản lý mạng truyền thống và phong cách quản lý mạng Tác tử di động lồng nhau. Kết quả cho thấy, so với phong cách quản lý truyền thống, quản lý mạng Tác tử di động lồng nhau linh hoạt hơn, có thể tái sử dụng, thông minh hơn và có thể phân phối được.

1.4 Tổng quan ứng dụng tác tử di động trong quản lý và bảo mật mạng cục bộ và đám mây

Belal Amro [24] nghiên cứu về ứng dụng Tác tử di động và phương thức tấn công và phòng thủ mạng và các bài báo nghiên cứu về an toàn thông tin, bảo mật cho SDN [10, 40, 41, 77, 94]. Bài báo nghiên cứu về phát hiện xâm nhập sử dụng Tác tử di động[28, 92] và cơ chế quản lý tài nguyên trong SDN [99].

Bài báo [19] của Al-Naymat, G, Al-kasassbeh, M, Al-Hawari khám phá tiềm năng của việc sử dụng dữ liệu SNMP-MIB để phát hiện các bất thường của mạng, đặc biệt là các cuộc tấn công từ chối dịch vụ (DoS), với sự trợ giúp của các kỹ thuật học máy. Các tác giả lập luận rằng các hệ thống phát hiện xâm nhập truyền thống phụ thuộc rất nhiều vào việc phân tích dữ liệu gói thô, có thể tốn kém về mặt tính toán và mất thời gian. Thay vào đó, họ đề xuất sử dụng SNMP-MIB, một cơ sở dữ liệu lưu trữ thông tin quản lý mạng được thu thập bởi Giao thức quản lý mạng đơn giản (SNMP).

Bài nghiên cứu của Almseidin và cộng sự [18] trình bày một phương pháp mới để phát hiện xâm nhập mạng bằng cách sử dụng các tham số SNMP-MIB và Nội suy quy tắc mờ (FRI). Bài báo bắt đầu bằng cách thảo luận về những thách thức liên quan đến các hệ thống phát hiện xâm nhập truyền thống, bao gồm nhu cầu xử lý dữ liệu mở rộng và thiếu ranh giới rõ ràng giữa các mẫu lưu lượng bình thường và bất thường. Sau đó, bài báo giới thiệu SNMP-MIB như một giải pháp tiềm năng, cung cấp thông tin phong phú về các thiết bị mạng mà không cần xử lý lưu lượng thô.

Qua phân tích các bài báo nghiên cứu trên, Luận án tập trung vào việc giải quyết hai vấn đề chính để xây dựng một giải pháp quản trị mạng cục bộ trong quá trình di chuyển lên mạng đám mây một cách an toàn, hiệu quả dựa trên công nghệ tác tử và đảm bảo an toàn thông tin cho hệ thống. Dưới đây là chi tiết về cách thức giải quyết hai vấn đề này:

Giải quyết vấn đề thứ nhất: Kết hợp các chuẩn quản lý mạng SNMP và CMIP

Vấn đề thứ nhất liên quan đến việc quản lý mạng cục bộ hiệu quả trong quá trình di chuyển lên đám mây. Để giải quyết vấn đề này, luận án đề xuất một phương pháp kiến trúc kết hợp hai chuẩn quản lý mạng phổ biến là SNMP (Simple Network Management Protocol) và CMIP (Common Management Information Protocol).

Bên cạnh đó, đề xuất xây dựng mô hình quản lý mạng dựa trên nền tảng tác tử di động đồng thời cũng đề xuất các cải tiến kỹ thuật cho nền tảng tác tử di động, bao gồm tối ưu hóa quy trình thực thi, nâng cao khả năng tương tác giữa các tác tử.

Giải quyết vấn đề thứ hai: Tăng cường tính an toàn của hệ thống mạng ứng dụng công nghệ tác tử

Vấn đề thứ hai liên quan đến việc đảm bảo tính an toàn cho các tác tử di động trong quá trình quản lý mạng. Công nghệ tác tử di động mang lại nhiều lợi ích như tính tự động hóa, khả năng di chuyển linh hoạt và xử lý phân tán. Tuy nhiên, các tác tử nguồn mở thường có những lỗ hổng bảo mật tiềm ẩn, dễ bị khai thác bởi các cuộc tấn công mạng nên cần cải thiện đảm bảo an toàn cho hệ thống tác tử.

Kết quả của giải pháp này là một hệ thống tác tử an toàn, có khả năng chống lại các cuộc tấn công mạng và đảm bảo tính toàn vẹn của dữ liệu trong quá trình quản lý mạng.

1.5 Kết luận vấn đề nghiên cứu

Thông qua nội dung tổng quan và đánh giá các hiện trạng nghiên cứu, tác giả nhận thấy một số nội dung cần nghiên cứu và đóng góp khoa học để nâng cao hiệu quả cho hệ thống mạng cục bộ và đám mây, đồng thời đảm bảo an toàn thông tin và bảo mật các vấn đề như sau:

Vấn đề 1: Đề xuất một phương pháp kiến trúc kết hợp hai chuẩn quản lý mạng phổ biến là SNMP và CMIP dựa trên nền tảng tác tử di động.

Vấn đề 2: Nâng cao tính an toàn bảo mật cho nền tảng tác tử di động và sử dụng nền tảng tác tử di động để đảm bảo an toàn cho các hệ thống thông tin.

CHƯƠNG 2 XÂY DỰNG MÔ HÌNH TÁC TỬ DI ĐỘNG QUẢN LÝ MẠNG TRÊN MẠNG CỤC BỘ VÀ ĐÁM MÂY

Để giải quyết vấn đề thứ nhất của bài toán dịch chuyển từ mạng cục bộ và đám mây, việc đề xuất giải pháp kết nối giữa hai giao thức quản lý mạng SNMP (Simple Network Management Protocol) và CMIP (Common Management Information Protocol), đồng thời xây dựng mô hình quản lý mạng cục bộ và đám mây trên nền tảng tác tử di động.

Bên cạnh đó, mô hình quản lý mạng cục bộ và đám mây trên nền tảng tác tử di động được đề xuất nhằm tận dụng khả năng tự động hóa, di chuyển linh hoạt và xử lý phân tán của công nghệ tác tử.

Do vậy, luận án đề xuất Xây dựng Mô hình Tác tử di động quản lý mạng trên mạng cục bộ và đám mây giúp quản lý hiệu quả các tài nguyên mạng, đồng thời đáp ứng nhu cầu mở rộng và tích hợp trong môi trường đa nền tảng.

2.1 Xây dựng Mô hình quản lý mạng cục bộ và đám mây dựa trên tác tử di động (CNMMA)

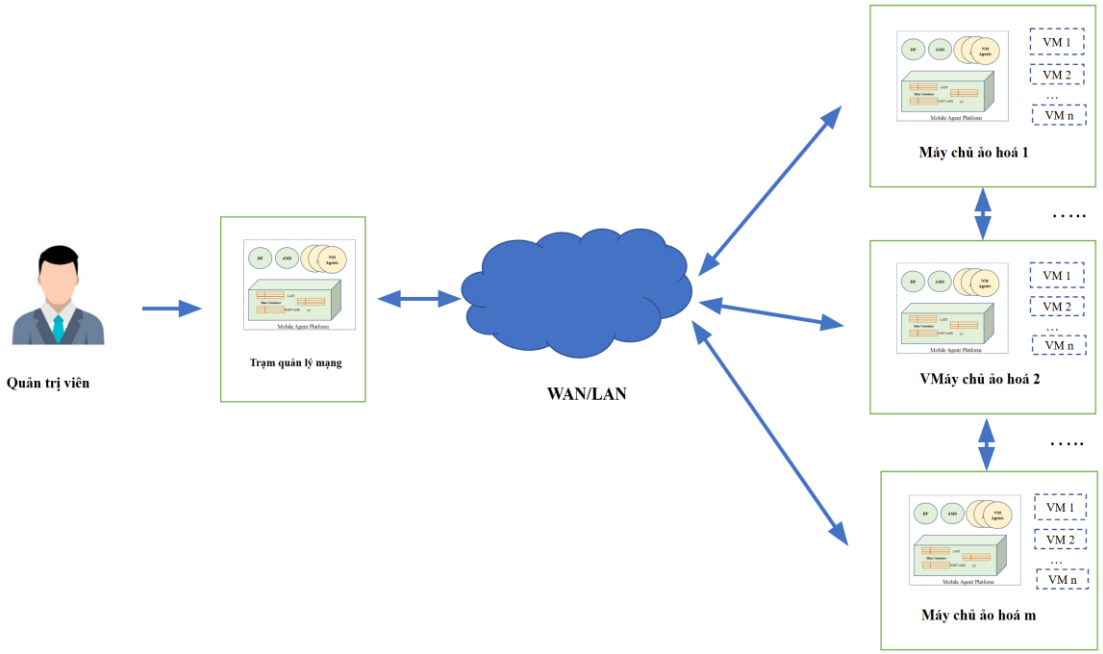
Trong chương 1 đã nghiên cứu tổng quan điểm yếu của các giao thức SNMP và CMIP trong giao tiếp quản lý mạng, vì vậy đề xuất xây dựng một mô hình quản lý mạng mới sẽ cố gắng khắc phục những vấn đề đó. Quản lý mạng thường yêu cầu sự hỗ trợ của một tác tử trong máy chủ được quản lý và cơ sở dữ liệu trong tác tử cung cấp thông tin quản lý cần thiết cho một ứng dụng quản lý, như trong mô hình dựa trên Tác tử di động là CNMMA (Cloud Network Management Mobile Agent).

2.2 Kiến trúc mô hình CNMMA

Mô hình CNMMA quản lý mạng đám mây dựa trên công nghệ Tác tử di động, trạm quản lý mạng, mối quan hệ quản lý máy chủ ảo hóa và kết hợp giữa giao thức quản lý mạng (giao thức CMIP) và đặc tả ACL (Ngôn ngữ tích lũy di động theo Đặc tả FIPA cho Tác tử di động). Các thành phần của Mô hình CNMMA bao gồm các thành phần cốt lõi cơ bản như trong Hình 2.1:

- Nền tảng tác tử di động
- Máy chủ quản lý mạng ảo hóa

- Trạm quản lý mạng (NMS)
- Quản lý mạng tác tử di động (NMMA)



Hình 2.1: Kiến trúc mô hình CNMMA

Các thành phần của mô hình CNMMA

1. Nền tảng tác tử di động (Mobile Agent Platform): là nền tảng cung cấp ứng dụng dựa trên Tác tử di động, một cơ sở hạ tầng để xây dựng và thực thi các tác tử di động phục vụ xử lý mạng. Nền tảng tác tử di động có ba chức năng cốt lõi bao gồm:

– *Quản lý phân cấp tác tử***:** Mỗi hệ thống Tác tử di động tương ứng với nút gốc của hệ thống phân cấp tác tử, được duy trì dưới dạng cấu trúc cây trong đó mỗi nút chứa một tác tử di động và các thuộc tính của nó. Di chuyển tác tử trong hệ thống phân cấp tác tử được thực hiện đơn giản như một sự biến đổi cấu trúc cây của hệ thống phân cấp [51].

– *Quản lý thực thi tác tử*: Mỗi tác tử có thể có nhiều hơn một luồng hoạt động dưới sự kiểm soát của hệ thống. Hệ thống duy trì trạng thái vòng đời của các tác tử. Khi trạng thái vòng đời của một tác tử bị thay đổi, ví dụ, khi tạo, chấm dứt hoặc di chuyển, hệ thống sẽ đưa ra một số sự kiện nhất định để gọi các phương pháp nhất định trong tác tử và các tác tử chứa của nó [51].

– *Quản lý bảo mật và tuần tự hóa tác tử*: Hệ thống có chức năng sắp xếp các tác

tử vào các luồng bit và giải mã chúng sau đó. Hệ thống xác minh xem một tác tử được sắp xếp có hợp lệ hay không để bảo vệ hệ thống chống lại các tác tử không hợp lệ hoặc độc hại, bằng cơ chế bảo mật [51].

2. Máy chủ quản lý mạng ảo hóa (Manager): được sử dụng để quản lý và giám sát toàn bộ mạng. Nó nhận được tất cả các thông tin và hiển thị nó. Nó có thể là một nhóm các máy chủ ảo hóa, có thể sử dụng các dịch vụ đám mây để có được Dịch vụ quản lý và giả định rằng Manager là nhóm máy chủ ảo hóa được lưu giữ trên đám mây. Trình quản lý cũng được cài đặt Nền tảng tác tử di động để tạo Mạng nền tác tử di động [88].

3. Trạm quản lý mạng (NMS): Một nút mạng chứa Nền tảng tác tử di động CNMMA để tạo môi trường cho phép Quản lý mạng Tác tử di động có thể chạy và quản lý trong đó.

4. Quản lý mạng Tác tử di động: là Tác tử di động được viết để hoạt động như tạo, di chuyển, xóa và thực hiện các chức năng quản lý mạng trong Nền tảng Tác tử di động.

2.3 Tích hợp giao thức SNMP và CMIP cho Tác tử di động trong quản lý mạng

2.3.1. Vấn đề với các chuẩn giao thức quản lý mạng

Trong quản lý mạng, để có chất lượng dịch vụ (QoS) tốt hơn, quản trị viên hệ thống mạng phải luôn nhận thức được trạng thái của các thiết bị mạng được gọi là tác tử, bao gồm tải CPU, bộ nhớ, mức sử dụng bộ nhớ, v.v. Hiện nay, giao thức SNMP (Simple Network Management Protocol) đã được sử dụng rộng rãi trong việc giám sát từ xa các thiết bị mạng và máy chủ. Ưu điểm chính của SNMP là sự đơn giản trong thiết kế. Ngoài ra, khi các thông điệp được truyền đạt giữa người quản lý và các thực thể cần quản lý thông qua các tác tử. Nhưng bảo mật thấp là điểm yếu chính của SNMP đã được cải thiện các phiên bản mới.

Mặt khác, để bù đắp cho các thiếu sót của giao thức SNMP, giao thức CMIP đã thiết kế để có thể được sử dụng cho các mạng lớn hơn và phức tạp hơn. Mô hình hướng đối tượng sẽ sử dụng để thiết kế và triển khai SNMP [41]. Ưu điểm chính của CMIP là khả năng xác định các kỹ thuật để bao gồm kiểm soát thủ công, bảo mật và

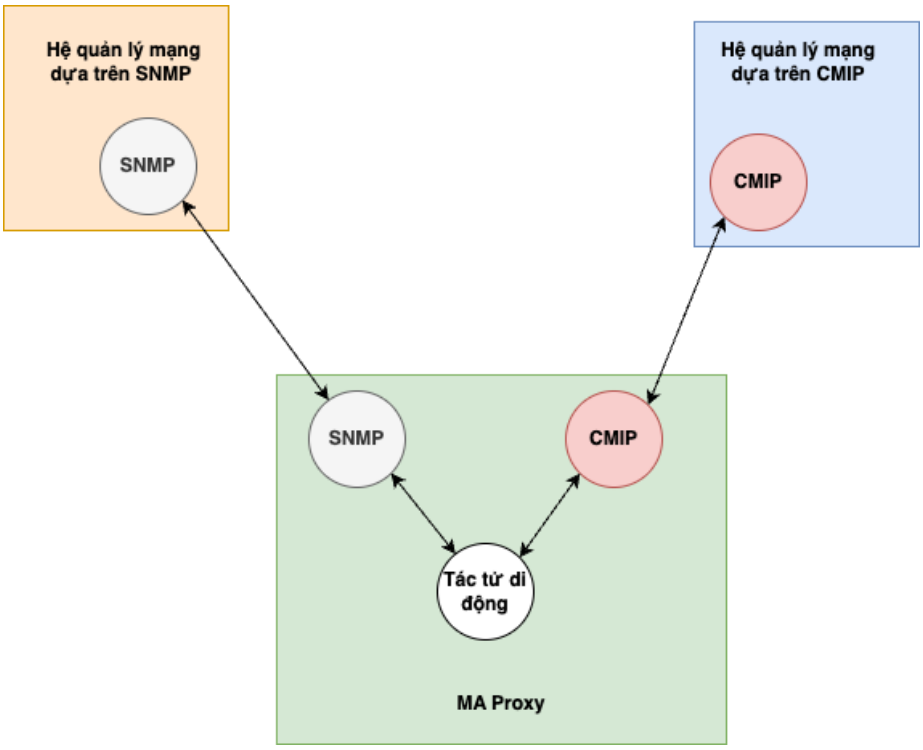
lọc thông tin quản lý. Nhược điểm của giao thức CMIP là thời gian chiếm dụng tài nguyên nhiều hơn SNMP [106].

Do vậy, cần thiết việc xây dựng một proxy cho tích hợp cả 2 bộ giao thức quản lý mạng SNMP và CMIP, đồng thời phép ánh xạ các tập lệnh giữa 2 bộ giao thức này để hệ thống quản lý mạng có thể quản trị được các loại thiết bị mạng khác nhau và kết nối các hệ thống quản trị với nhau.

2.3.2. Tác tử di động proxy cho quản lý mạng

Tác tử di động proxy MA (Mobile Agent proxy) được đề xuất cung cấp mô phỏng các dịch vụ CMIS bằng cách ánh xạ chúng đến các thông điệp SNMP tương ứng. Nó cho phép quản lý các đối tượng Internet MIB-II bằng trình quản lý CMIP hỗ trợ giao thức quản lý mạng CMIP và các dịch vụ CMIS.

Proxy MA mô phỏng các dịch vụ như xác định phạm vi và lọc, xử lý các hoạt động CMIS và dịch các bẫy SNMP sang thông báo CMIS. Hình 2.2 cho thấy sơ đồ luồng dữ liệu ngữ cảnh của tác tử proxy CMIP/SNMP.



Hình 2.2: *Tác tử di động proxy để quản lý mạng*

Proxy MA thực hiện các chức năng sau:

- Quản lý thiết lập/giải phóng kết nối với trình quản lý CMIP.

- Truyền dữ liệu giữa người quản lý CMIP và tác tử Internet bao gồm:

+ chuyển các đơn vị dữ liệu giao thức phản hồi và chỉ báo CMIP (PDU) với trình quản lý CMIP.

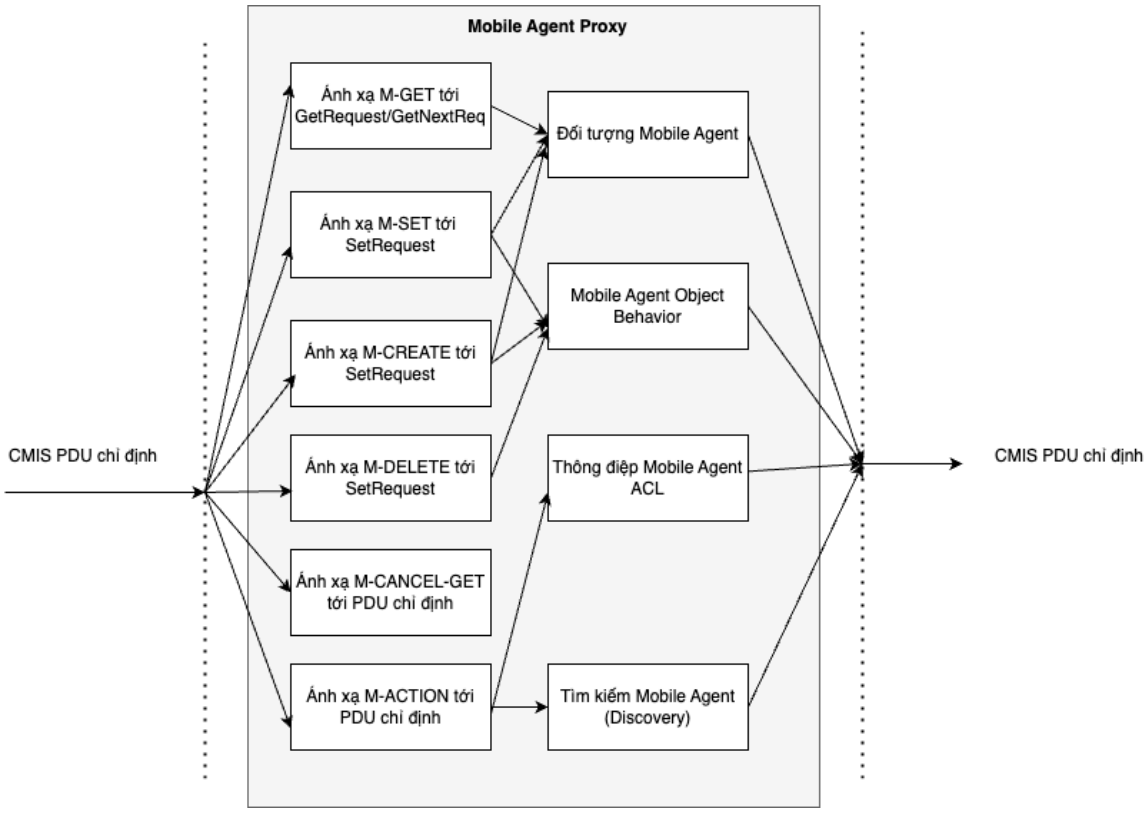
+ chuyển yêu cầu SNMP và tin nhắn phản hồi với đại lý Internet.

- Chức năng mô phỏng dịch vụ proxy như:

+ Ánh xạ CMIS sang SNMP

+ Ánh xạ SNMP sang CMIS

Proxy MA duy trì một tệp cấu hình trong kho dữ liệu chung để bảo toàn thông tin trong quá trình truyền tin nhắn. Hình dưới cho thấy sơ đồ luồng dữ liệu cho ánh xạ dịch vụ này.



Hình 2.3*: Ánh xạ dịch vụ CMIP và SNMP thông qua Tác tử di động*

2.4 Bảng ánh xạ giao thức quản lý mạng CMIP và SNMP

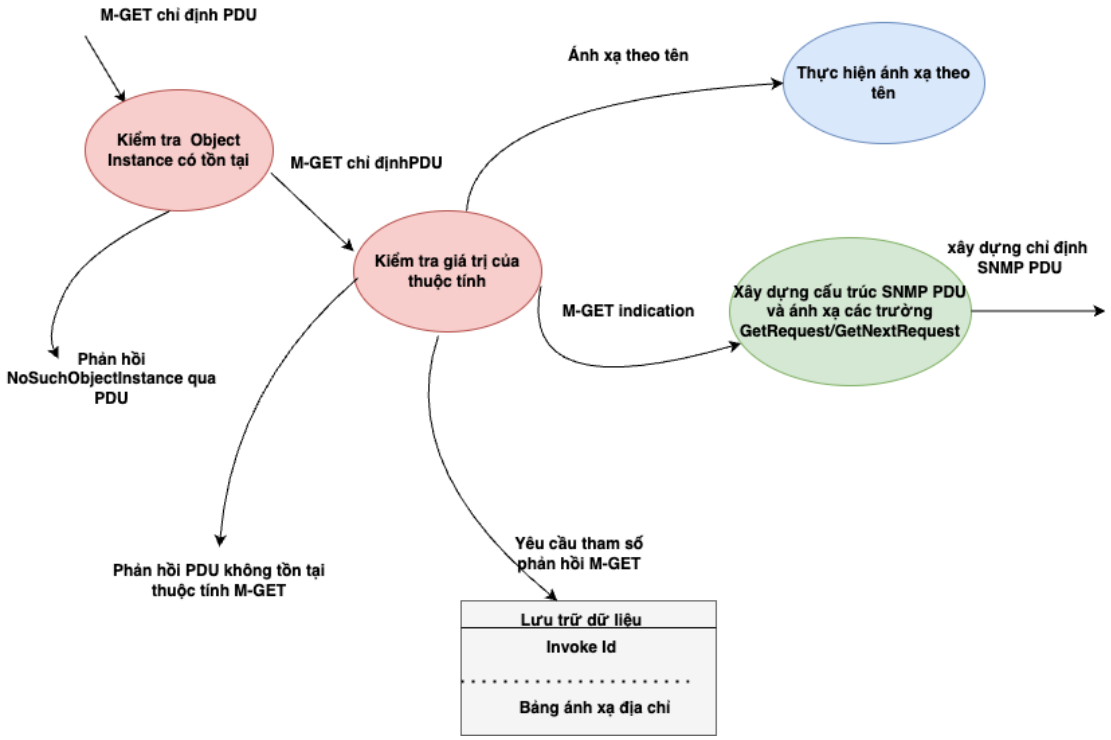
2.4.1 M-GET to GetRequestI GetNextRequest Mapping

Khi nhận được yêu cầu CMIS M-GET, proxy trước tiên xác minh sự tồn tại của đối tượng được quản lý cơ sở. Nếu đối tượng cơ sở được chỉ định trong yêu cầu không

tồn tại trong tác tử Internet, thì proxy sẽ gửi phản hồi lỗi CMIS "NoSuchObje ctlnstance" trở lại CMIPmanager [63].

Tham số attributeldlist của CMIP PDU là một tùy chọn service-user. Nếu nó trống, điều đó có nghĩa là proxy phải truy xuất giá trị của tất cả các thuộc tính được chỉ định trong đối tượng được quản lý cơ sở theo định nghĩa mẫu GDMO [90].

Nếu thuộc tính yêu cầu CMIS không trống, proxy sẽ xác minh (các) thuộc tính được chỉ định cho lớp đối tượng cơ sở. Nếu nó không được xác định trong lớp đối tượng cơ sở, proxy trả về lỗi CMIS "noSuchAttribute" và không gửi yêu cầu SNMP đến tác tử Internet.



Hình 2.4: Luồng dữ liệu để ánh xạ dịch vụ M-GET tới SNMP GetRequest/ GetNextRequest.

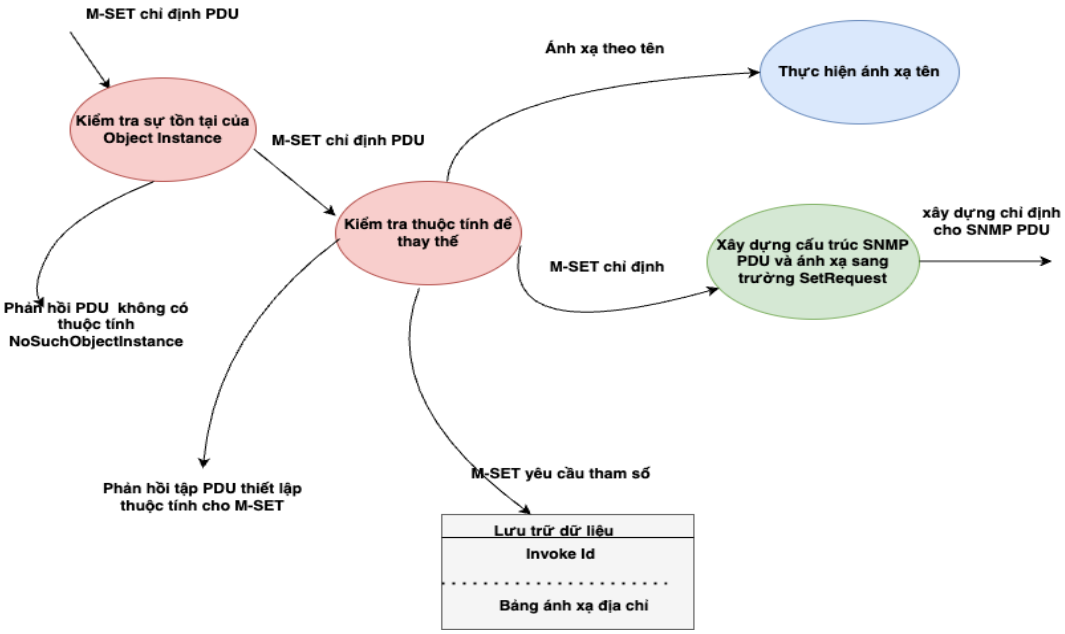
Sau khi xác thực thành công PDU chỉ báo M-GET, proxy sẽ xây dựng định dạng tin nhắn SNMP với loại PDU được đặt thành SNMP GetRequest hoặc GetNextRequest. Do đó, proxy đã ánh xạ PDU chỉ báo CMIP M-GET đến với thông báo SNMP GetRequest / GetNextRequest. Tuy nhiên, thông báo này được xây dựng một phần vì nó yêu cầu các trường khác của thư phải được ánh xạ thích hợp. Điều này đạt được bởi proxy bằng cách thực hiện chức năng ánh xạ tên, cung cấp danh

sách ràng buộc biến. Id yêu cầu, địa chỉ Nguồn và Đích được trích xuất từ bảng dịch địa chỉ [63].

2.4.2 M-SET thành SetRequest Mapping

Khi nhận được yêu cầu CMIS M-SET, proxy trước tiên sẽ kiểm tra xem đối tượng được quản lý cơ sở được chỉ định có tồn tại hay không. Nếu không tìm thấy đối tượng cơ sở trong tác tử Internet, proxy sẽ gửi phản hồi lỗi CMIS "NoSuchObjectInstance" tới trình quản lý CMIP [63].

Đối với mỗi phiên bản đối tượng được chọn trong chỉ báo CMIP M-SET, proxy sẽ tạo một hoặc nhiều thông báo SNMP SetRequest để cập nhật các thuộc tính được xác định bởi tham số CMIS modificationList, theo toán tử sửa đổi được chỉ định. Proxy chỉ hỗ trợ toán tử sửa đổi "thay thế"; nếu không có toán tử nào được chỉ định trong yêu cầu CMIS, "thay thế" được giả định theo mặc định. Proxy tham khảo lược đồ MIB của nó để xác nhận rằng thuộc tính đang được sửa đổi có thuộc tính REPLACE cần thiết cho mô phỏng CMIS M-SET. Nếu toán tử sửa đổi không được hỗ trợ được bao gồm trong yêu cầu M-SET, proxy sẽ trả về lỗi CMIS "Lỗi danh sách đặt" cho trình quản lý CMIP [63].



Hình 2.5: Luồng dữ liệu để ánh xạ dịch vụ M-SET tới SNMP SetRequest.

2.4.3 M-CREATE đến SetRequest Mapping

Proxy được giao nhiệm vụ xác thực từng chỉ báo CMIS M-CREATE đến. Nó kiểm tra xem một phiên bản có thể được tạo dựa trên mệnh đề CREATE trong các mẫu NAME BINDING được chỉ định cho lớp đối tượng hay không. Nếu không được phép tạo, nó sẽ trả về phản hồi lỗi CMIS "classInstance-Conflict" [98, 63].

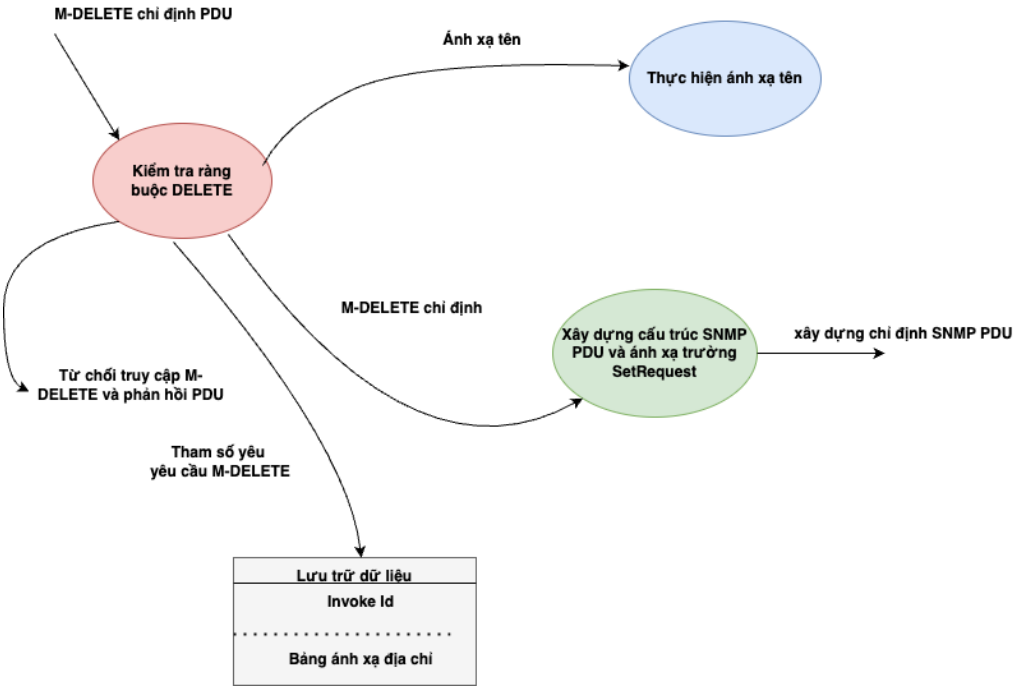
Proxy đánh giá thêm xem phiên bản được yêu cầu có thể được tạo theo phiên bản đối tượng cao cấp được chỉ định trong yêu cầu M-CREATE hay không, như được định nghĩa bởi mẫu LIÊN KẾT TÊN. Nếu mẫu NAME BINDING không hỗ trợ mối quan hệ ngăn chặn được chỉ định, nó sẽ trả về lỗi CMIS "invalidObjectInstance". Ngoài ra, nếu phiên bản đối tượng đã tồn tại, lỗi CMIS "phiên bản đối tượng được quản lý trùng lặp" sẽ được phát hành [84, 106].

Nếu chỉ báo CMIS M-CREATE bao gồm một đối tượng tham chiếu, proxy sẽ truy xuất phiên bản được tham chiếu từ tác tử được quản lý Internet. Để có được các giá trị đối tượng tham chiếu, nó sẽ gửi một SNMP GetNextRequest. Nếu đối tượng tham chiếu bị thiếu, proxy sẽ gửi lỗi CMIS "Không có đối tượng tham chiếu như vậy" đến trình quản lý CMIP.

2.4.4 M-DELETE đến SetRequest Mapping

Proxy xác định từ các mẫu LIÊN KẾT TÊN nếu phiên bản được chỉ định trong yêu cầu có thể bị xóa. Nếu LIÊN KẾT TÊN không cho phép xóa đối tượng đã xác định, lỗi CMIS "Truy cập bị từ chối" sẽ được trả về.

Nếu phiên bản đối tượng được xác định trong yêu cầu CMIS M-DELETE tồn tại, thao tác xóa được thực hiện. Việc xóa một đối tượng được thực hiện bằng cách đặt đối tượng cột trạng thái thành một giá trị không hợp lệ. Xóa đối tượng đạt được bằng cách gửi SNMP SetRequest để thay đổi giá trị rowStatus thành "phá hủy" [85].



Hình 2.6: Luồng dữ liệu để ánh xạ dịch vụ M-DELETE tới SNMP

2.5 Xử lý yêu cầu CMISE khác

Tác tử proxy xử lý các yêu cầu M-CANCEL-GET và M-ACTION như sau:

2.5.1 Dịch vụ M-CANCEL-GET

Tác tử proxy không cần gửi bất kỳ yêu cầu SNMP nào để mô phỏng yêu cầu CMIS M-CANCEL-GET. Khi nhận được chỉ báo M-CANCEL-GET, proxy sẽ ngừng gửi bất kỳ phản hồi CMIS M-GET nào khác đến trình quản lý CMIP cho yêu cầu đã hủy cụ thể đó. Ngoài ra, khi nhận được chỉ báo M-CANCEL-GET, proxy sẽ ngừng tất cả SNMP GetRequests trong tương lai cho tác tử Internet liên quan đến yêu cầu M-GET bị hủy. Nếu tác tử Internet vẫn trả về GetResponses cho yêu cầu đã hủy, proxy sẽ loại bỏ các phản hồi này [85].

PDU chỉ báo M-CANCEL-GET bao gồm mã định danh Gọi của yêu cầu M-GET sẽ bị hủy. Nếu proxy không nhận ra mã định danh Invoke, nó sẽ trả về lỗi CMIS "no such invoke identifier" cho trình quản lý CMIP [63, 85].

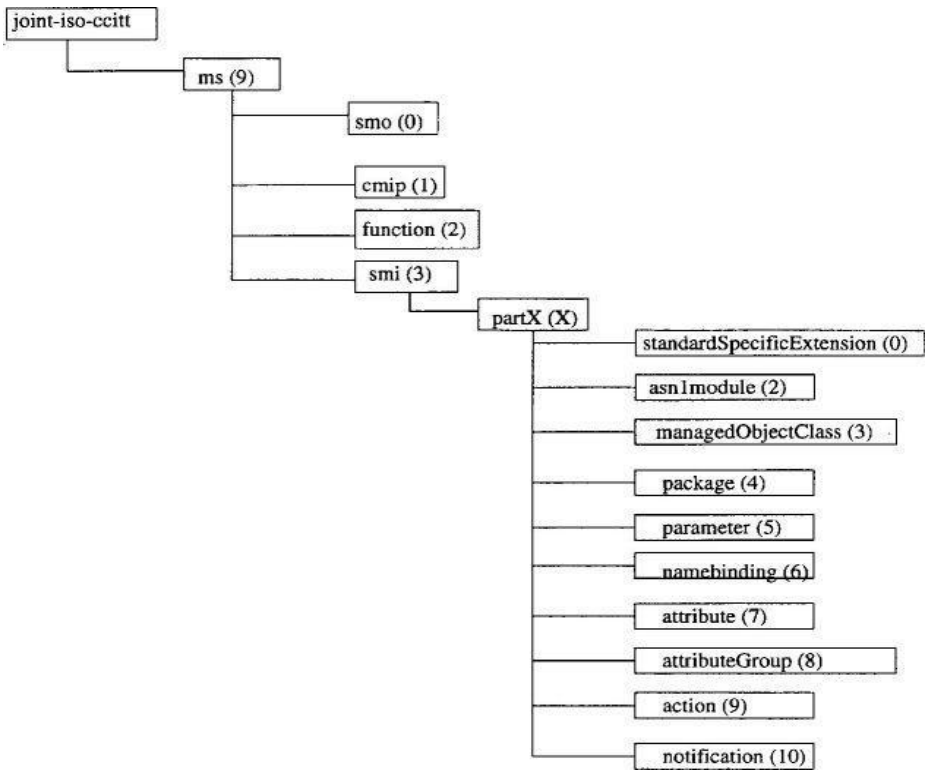
2.5.2 M-ACTION Service

Vì MIB Internet thiếu các hành động được xác định, hoạt động CMIS M-ACTION không tương ứng với bất kỳ hoạt động SNMP nào. Do đó, proxy không xử

lý thao tác này; khi nhận được chỉ báo M-ACTION, nó chỉ trả về phản hồi lỗi CMIS "Hoạt động không được nhận dạng".

2.6 Đăng ký và đặt tên

Đăng ký đảm bảo tính duy nhất của các loại phần tử thông tin quản lý, trong khi đặt tên cung cấp một cách để phân biệt các trường hợp của từng loại và định vị chúng một cách chính xác trong MIB. Xác định các lớp đối tượng được quản lý liên quan đến việc đăng ký các mã định danh duy nhất trên toàn cầu, được gọi là mã định danh đối tượng, đại diện cho các yếu tố khác nhau của lớp đối tượng được quản lý, chẳng hạn như tên, loại thuộc tính và hơn thế nữa. Các mã định danh này rất cần thiết trong các giao thức quản lý để xác định duy nhất các khía cạnh khác nhau của các đối tượng được quản lý, bao gồm các thuộc tính, hoạt động và thông báo của chúng. Tất cả các nhóm Internet và bảng khái niệm được dịch thành các lớp đối tượng được quản lý CMIP, với các đối tượng cột và vô hướng trở thành thuộc tính trong các lớp đối tượng này. Ngoài ra, các ràng buộc tên cho các trường hợp đối tượng định vị trong hệ thống phân cấp đặt tên cũng phải được đăng ký [106].



Hình 2.7: Đăng ký theo cây con ISO/CCITT chung [106].

2.7 Ánh xạ tên từ ISO/CCITT sang Internet

Lựa chọn lớp đối tượng được quản lý được sử dụng để xác định một tập hợp các trường hợp đối tượng được quản lý trong cây thông tin quản lý mà trên đó yêu cầu CMIS được áp dụng. Điều này được thực hiện bởi hoạt động xác định phạm vi CMIS. Phạm vi được sử dụng để chọn các phiên bản đối tượng ứng viên trong cây thông tin quản lý (MIT) mà các hoạt động có thể áp dụng. Trước khi OID phiên bản đối tượng có thể được dịch từ cây thông tin quản lý CMIP, cần phải lấy tập hợp các đối tượng được quản lý nằm trong phạm vi [69].

Khi nhận được yêu cầu từ trình quản lý CMIP, proxy trước tiên xác định tập hợp các lớp đối tượng nằm trong phạm vi. Thông tin liên quan đến các lớp đối tượng cấp trên và cấp dưới có trong mẫu ràng buộc tên được sử dụng để đi qua cây trong quá trình xác định phạm vi. Nếu tham số scope không có trong yêu cầu, phạm vi mặc định là chính đối tượng cơ sở. Lớp đối tượng cũng chứa tên thuộc tính được chỉ định là tham số filter [63, 106].

Các lớp đối tượng có phạm vi đã chọn và nắm tay định danh thuộc tính được cung cấp bởi yêu cầu CMIS được proxy sử dụng để trích xuất các OID đăng ký ISO / CCITT từ các MIB GDMO. Như đã thảo luận, các OID lớp đối tượng CMIP trước đó và các OID thuộc tính CMIP tuân theo mô hình chung. Do đó, bằng cách loại bỏ các OID được liên kết với MOG và ATT trong cây đăng ký CMIP, chúng ta có thể nhận được các OID Internet. Để có được tên đối tượng Internet, chúng ta cần nối thêm đối tượng internet OID với thuộc tính đặt tên được liên kết với nó. Thuộc tính đặt tên được trích xuất từ tên phân biệt (DN) được cung cấp cùng với bất kỳ yêu cầu CMIS nào [90,106].

2.7.1 Dịch lỗi SNMP sang CMIS

Tác tử proxy dịch các phản hồi lỗi SNMP đến thành phản hồi lỗi CMIS được gửi đến trình quản lý CMIP [90]. Bảng 2.1 sau mô tả ánh xạ lỗi này.

2.7.2 Hoạt động của bộ lọc

Bộ lọc là một biểu thức Boolean bao gồm một hoặc nhiều khẳng định liên quan đến sự hiện diện hoặc giá trị của các thuộc tính trong đối tượng được quản lý trong phạm vi. Proxy áp dụng bộ lọc này cho tất cả các phiên bản đối tượng nhận được từ

tác tử được quản lý, chỉ chọn những phiên bản đáp ứng tiêu chí lọc. Sau đó, gửi PDU phản hồi CMIS đã lọc đến trình quản lý CMIP [63, 90].

Bảng 2.1: Bản dịch lỗi SNMP sang CMIS

Lỗi SNMP	Lỗi CMIS đã chuyển đổi
noError	Giới hạn độ phức tạp
noSuchName	Không có đối tượng như vậy
badValue	Xử lý thất bại
readonly	Xử lý thất bại
genErr	Xử lý thất bại
noAccess	Không có thuộc tính như vậy
wrongType	Giá trị thuộc tính không hợp lệ
wrongLength	Xử lý thất bại
wrongEncoding	Xử lý thất bại
wrong Value	Giá trị thuộc tính không hợp lệ
noCreation	Phiên bản đối tượng không hợp lệ
inconsistentValue	Giá trị thuộc tính không hợp lệ
resourceUnavailable	Giới hạn tài nguyên
commitFailed	Xử lý thất bại
undoFailed	Xử lý thất bại
authorizationError	Truy cập bị từ chối
notWritable	Truy cập bị từ chối
inconsistentName	Đối số bị nhập sai

2.8 Ánh xạ SNMP Trap tới CMIS M-EVENT-REPORT

Macro 'NOTIFICATION-TYPE' xác định dữ liệu mà tác tử SNMP gửi khi một sự kiện đặc biệt xảy ra, được đóng gói dưới dạng PDU Trap. Khi nhận được PDU Trap, tác tử proxy sẽ chuyển tiếp thông tin đến người quản lý CMIP ISO / CCITT. Nó sử dụng các tham số Trap PDU để xây dựng PDU yêu cầu CMIS 'M-EVENT-REPORT', trong đó tất cả các bẫy SNMP được ánh xạ tới một loại sự kiện thông báo CMIS chung có tên là 'InternetAlarm' [63, 106].

Mệnh đề "Mô tả" trong macro trở thành thông số Thông tin sự kiện, cung cấp mô tả chi tiết về sự kiện. Ngoài ra, SNMP Trap PDU bao gồm danh sách ràng buộc biến, trong đó biến đầu tiên, 'sysUpTime.0', được sử dụng để đặt tham số Thời gian sự kiện trong yêu cầu 'M-EVENT-REPORT' [106].

2.9 Phân tích làm rõ hơn độ tương thích ngữ nghĩa giữa CMIP và SNMP

Một bảng ánh xạ CMIP-SNMP đơn thuần chỉ liệt kê "CMIP attribute X tương ứng với SNMP MIB object Y" thường chưa đủ, bởi sự khác biệt nền tảng về mô hình quản lý mạng (thủ tục vs. hướng đối tượng, có state vs. stateless) dẫn đến nhiều vấn đề về tương thích ngữ nghĩa.

Do vậy, cần bổ sung các khía cạnh để làm rõ phân tích khi ánh xạ theo Bảng 2.2 như sau:

Bảng 2.2: Bảng câu hỏi phân tích ánh xạ ngữ nghĩa giữa CMIP và SNMP

Khía cạnh	**Câu hỏi phân tích**
Ngữ nghĩa chức năng	Attribute trong CMIP là thông tin cấu hình hay trạng thái? SNMP có biến tương ứng không?
Kiểu dữ liệu	CMIP dùng ASN.1 (có thể là SET OF, SEQUENCE, CHOICE) – SNMP dùng SMIv2 (đơn giản hơn, không có cấu trúc lồng sâu). Có mất mát cấu trúc không?
Hành vi đọc/ghi	CMIP phân biệt rõ read, write, create, delete, action. SNMP chỉ có get/set trên OID. Hành vi nào bị mất?
Phạm vi giá trị	CMIP cho phép ràng buộc phức tạp; SNMP chỉ có range/enum. Có giá trị hợp lệ trong CMIP nhưng nằm ngoài phạm vi SNMP không?

Khía cạnh	Câu hỏi phân tích
Sự tồn tại	CMIP: attribute có thể bắt buộc/tùy chọn/điều kiện. SNMP: object có thể Mandatory/Optional/Obsolete. Có ánh xạ sai lệch không?
Thông báo (Notification)	CMIP dùng event report (có tham số phức tạp). SNMP dùng trap/inform (cố định hoặc mở rộng). Có giữ được toàn bộ ngữ cảnh sự kiện không?

Dựa trên bảng câu hỏi phân tích các khía cảnh, việc bổ sung bảng ánh xạ cho các thuộc tính và trạng thái ngữ nghĩa như sau:

Bảng 2.3: Bảng bổ sung độ tương thích và phân tích ánh xạ ngữ nghĩa giữa CMIP và SNMP

CMIP	SNMP	Độ tương thích	Phân tích ngữ nghĩa
operationalState (enumerated: disabled, enabled, testing)	ifOperStatus (INTEGER: up, down, testing, dormant, notPresent, lowerLayerDown)	Một phần (Partial)	Cả hai đều mô tả trạng thái hoạt động của một đối tượng mạng. Khác biệt: CMIP không có các trạng thái dormant, notPresent, lowerLayerDown → ánh xạ cần gom nhóm. disabled trong CMIP có thể tương ứng với down nhưng

CMIP	SNMP	Độ tương thích	Phân tích ngữ nghĩa
			mất thông tin lý do.
Hành vi m-create trên MOI (Managed Object Instance)	RowStatus TEX TUAL CONVENTION (createAndGo, createAndWait, destroy)	Cần biến đổi (Transformable)	CMIP tách biệt tạo đối tượng và khởi tạo attribute. SNMP gộp vào một biến đặc biệt (RowStatus). Khác biệt: CMIP hỗ trợ giao dịch (transaction) khi tạo; SNMP không có tính nguyên tử cho nhiều bước → khi ánh xạ cần thêm thứ tự thao tác set.
alarmRecord (gồm: cause, perceivedSeverity, probableCause, additionalText, ...)	Một hàng trong alarmTabl e (VD: RMON2 alarm hoặc NOTIFICATIO N-LOG-MIB)	Một phần (Partial)	CMIP có cấu trúc alarmRecord pho ng phú với nhiều tham số tùy chọn, bao gồm cả tham số về hành động khắc phục (proposedRepairActio ns). SNMP không có cấu trúc tương đương chuẩn cho toàn bộ – chỉ có thể dùng kết hợp nhiều bảng và log. Kết luận: Mất

CMIP	SNMP	Độ tương thích	Phân tích ngữ nghĩa
			thông tin về đề xuất sửa chữa.
operationalState của lớp top	ifOperStatus (IF-MIB)	Một phần (Partial)	Ánh xạ disabled → down, bỏ qua dormant nếu không dùng
m-create action	RowStatus = createAndGo	Cần biến đổi (Transformable)	Cần set nhiều bước trong SNMP, kiểm tra RowStatus cuối cùng

2.10 Xây dựng mô hình tác tử di động Quản lý mạng

Để chứng minh mô hình của Quản lý mạng Tác tử di động đang hoạt động, việc thực hiện thử nghiệm trong phòng thí nghiệm trên mô hình thực tế bằng cách sử dụng Nền tảng tác tử di động nguồn mở để tạo ứng dụng quản lý mạng, triển khai giao thức và cơ sở hạ tầng mạng để mô phỏng.

Với mô hình kết nối mạng truyền thống, việc trao đổi dữ liệu, thông điệp giữa hai phần mềm cần phải kết nối liên tục và yêu cầu hệ thống mạng ổn định, băng thông và độ trễ đáp ứng yêu cầu của phần mềm ứng dụng [64].

Kiến trúc nền tảng tác tử

Nền tảng tác tử di động bao gồm hai thành phần: Tác tử di động và hệ thống thời gian thực.

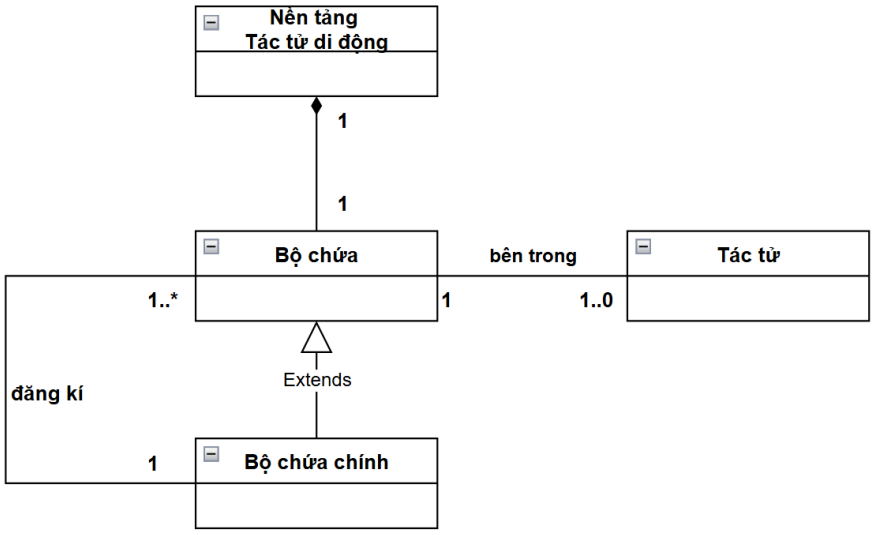
Tác tử di động: định nghĩa hành vi của tác tử phần mềm. Thành phần còn lại còn được gọi là nền tảng tác tử, hệ thống tác tử, máy chủ tác tử, và hỗ trợ thực thi và dịch chuyển. Cần một kiến trúc tương tự tồn tại trên các máy tính mà tác tử có

thể di chuyển tới, khi đó mỗi tác tử di động chạy trong môi trường thời gian thực tại mỗi máy tính nó đang cư trú. Khi tác tử yêu cầu tới hệ thống thời gian thực để tự dịch chuyển, hệ thống thời gian thực có thể di chuyển tác tử tới hệ thống thời gian thực tại máy tính đích, mang theo trạng thái và mã nguồn cùng với nó. Mỗi hệ thống thời gian thực chạy bên trên môi trường hệ điều hành như một phần mềm lớp giữa (middleware).

2.10.1. Nền tảng tác tử di động

Nền tảng tác tử di động là một khung nền tảng Tác tử di động, bao gồm các container tác tử có thể được phân phối qua mạng. Các tác tử sống trong các container là tiến trình Java cung cấp thời gian thực chạy trên nền tảng và tất cả các dịch vụ cần thiết để lưu trữ và thực thi các tác tử trong Hình 2.8. có một container đặc biệt, được gọi là container chính, đại diện cho điểm bootstrap của một nền tảng: nó là container đầu tiên được khởi chạy và tất cả các container khác phải tham gia vào một container chính bằng cách đăng ký với nó.

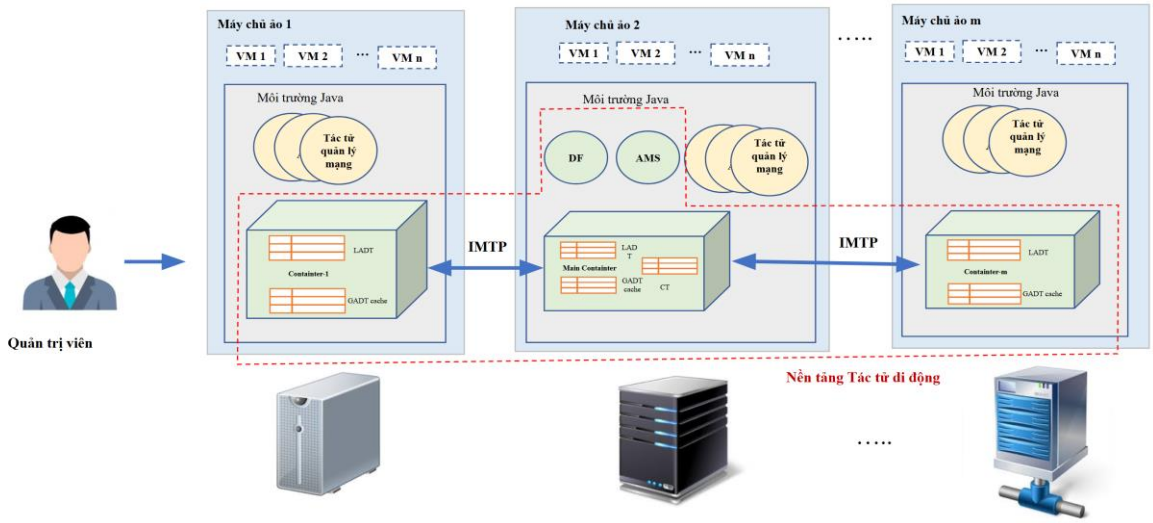
Sơ đồ UML trong Hình 2.8 sau sơ đồ hóa mối quan hệ giữa các yếu tố kiến trúc chính của nền tảng tác tử di động.



Hình 2.8: Mối quan hệ giữa các yếu tố thành phần kiến trúc nền tảng tác tử di động

Để triển khai mô hình CNMMA (Cloud Network Management Mobile Agent), đề xuất cơ sở kiến trúc thời gian chạy trên nền tảng tác tử di động như hình Triển

khai mô hình CNMMA dựa trên khung nền tảng tác tử di động như Hình 2.9.



Hình 2.9: Triển khai mô hình CNMMA dựa trên khung nền tảng tác tử di động

2.10.2. ASN.1 cho giao thức CMIP

Việc nghiên cứu định nghĩa cú pháp trừu tượng ASN.1 cho giao thức CMIP và tạo ra các chức năng cơ bản của giao thức CMIP như: M-Get, M-Set, M-Action, M-Create, M-Delete, M-Event-Report dựa trên ITU-T X.10 (ISO / IEC 9595: 1998) [90].

Bảng 2.4: Mã giả Psedocode ASN.1 cho giao thức CMIP

```
-- Hoạt động CMISE
- Hoạt động hành động (M-ACTION)
m-Action OPERATION ::= {
   ARGUMENT ActionArgument
   TRẢ KẾT QUẢ FALSE
   LUÔN TRẢ LỜI SAI
   CODE địa phương:6
}
m-Action-Xác nhận OPERATION::= {
   ARGUMENT ActionArgument
   KẾT QUẢ Hành động
   OPTIONAL TRUE - kết quả này là có điều kiện;
   - đối với các điều kiện xem 8.3.3.2.9 của ITU-T Rec. X.710
```

```
        LỖI
            {accessDenied | classInstanceConflict | complexityLimitation | invalidScope
                | invalidArgumentValue | invalidFilter | noSuchHành động |
noSuchArgument |
                noSuchObjectClass | noSuchObjectInstance | xử lýThất bại |
                syncNotSupported}
        LINKED {m-linked-reply}
        CODE địa phương:7
    }
```

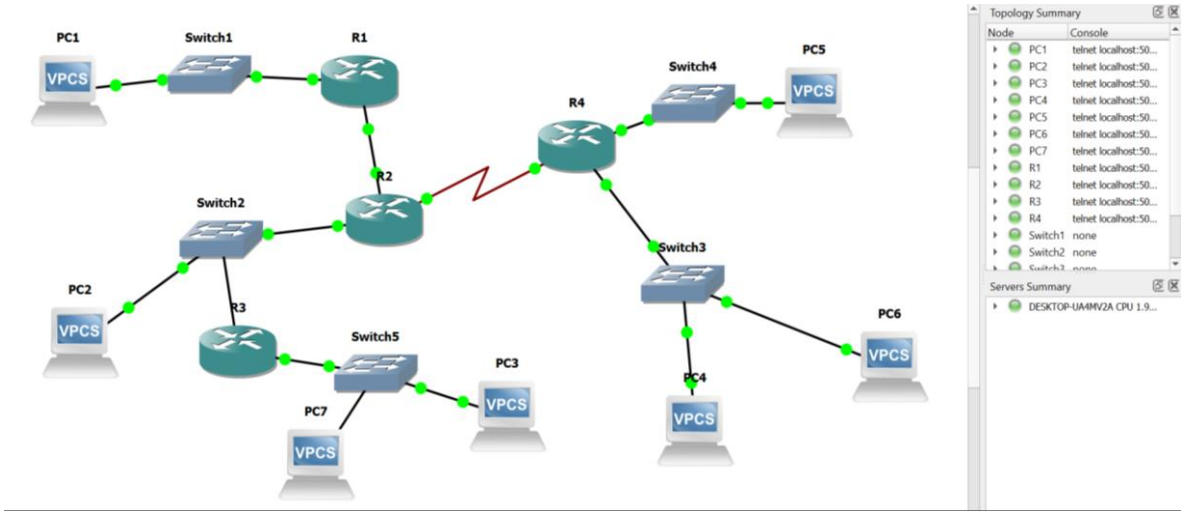
2.10.3. ACL trong nền tảng tác tử di động

Giao tiếp tác tử có lẽ là tính năng cơ bản nhất của nền tảng tác tử di động và được thực hiện theo các thông số kỹ thuật của FIPA. Mô hình giao tiếp dựa trên việc truyền thông điệp không đồng bộ. Định dạng cụ thể của tin nhắn trong nền tảng tác tử di động tuân thủ định dạng được xác định bởi cấu trúc tin nhắn FIPA-ACL [40]. Mỗi thư bao gồm các trường sau [41]:

- Người gửi thư.
- Danh sách người nhận.
- Hành động giao tiếp (còn được gọi là 'biểu diễn') cho biết người gửi dự định đạt được điều gì bằng cách gửi tin nhắn.
- Nội dung chứa thông tin thực tế cần trao đổi bằng tin nhắn.
- Ngôn ngữ nội dung cho biết cú pháp được sử dụng để thể hiện nội dung. Cả người gửi và người nhận phải có khả năng mã hóa và phân tích cú pháp các biểu thức tuân thủ cú pháp này để giao tiếp có hiệu quả.

2.10.4. Trình mô phỏng quản lý mạng

Để thử nghiệm nguyên mẫu giao thức CMIP và lý thuyết về Mô hình CNMMA dẫn tới một trình mô phỏng mạng dựa trên GNS3 và tạo kết nối từ mô phỏng nối mạng đến PC ảo hóa vật lý cài đặt khung nền tảng tác tử di động để thử nghiệm tạo và di chuyển tác tử Quản lý mạng trên nền tảng Tác tử di động.



Hình 2.10: Trình mô phỏng quản lý mạng bằng GNS3

2.10.5 Xây dựng luồng thực nghiệm ánh xạ từ CMIP - SNMP

Thực nghiệm 1 – Ánh xạ CMIP → SNMP (M-GET Mapping)

Mục tiêu

Kiểm chứng quá trình ánh xạ từ yêu cầu CMIP M-GET sang SNMP GET, đảm bảo khả năng chuyển đổi chính xác các đối tượng CMIP (objectClass, attributes) sang các OID SNMP, thực thi truy vấn và nhận phản hồi chính xác.

Bảng 2.5: Bảng ánh xạ Object Class

CMIP Object Class	**SNMP Base OID**	**MIB Group**
system	1.3.6.1.2.1.1	SNMP-SYSTEM-MIB
interface	1.3.6.1.2.1.2	IF-MIB
ip	1.3.6.1.2.1.4	IP-MIB
tcp	1.3.6.1.2.1.6	TCP-MIB

Bảng 2.6: Bảng ánh xạ Attribute

CMIP Attribute	**SNMP OID**	**MIB Object**
description	1.3.6.1.2.1.1.1.0	sysDescr
uptime	1.3.6.1.2.1.1.3.0	sysUpTime
contact	1.3.6.1.2.1.1.4.0	sysContact

Cách cài đặt

- Môi trường: Ubuntu 22.04, Java 21, JADE 5.1.0 Framework.
- Hệ thống: CMIP Agent, SNMP Agent, Protocol Mapping Engine.
- Quy trình: Nhận yêu cầu CMIP M-GET → Ánh xạ Object Class → Ánh xạ Attributes → Thực thi SNMP GET → Chuyển đổi phản hồi SNMP thành định dạng CMIP.

Số liệu

Trong 20 lần thử nghiệm cho thấy thời gian trung bình khoảng 50.2 ms cho mỗi yêu cầu M-GET.

Phân tích

Thời gian xử lý ổn định trong khoảng 47–54 ms. Mỗi yêu cầu M-GET thực thi đồng thời ba truy vấn SNMP, kết quả ánh xạ chính xác và phản hồi đúng định dạng CMIP.

Kết luận

Ánh xạ CMIP → SNMP hoạt động chính xác và ổn định, phù hợp triển khai trong hệ thống quản lý mạng tập trung hoặc phân tán.

Thực nghiệm 2 – Ánh xạ SNMP → CMIP (GET Mapping)

Mục tiêu

Xác minh khả năng ánh xạ ngược từ SNMP GET sang CMIP M-GET, đảm bảo nhận diện OID và tạo phản hồi CMIP hợp lệ.

Cách cài đặt

- SNMP Request gửi từ thiết bị bộ định tuyến1 đến Gateway.
- Gateway phân tích OID, tìm Object Class và thuộc tính tương ứng.
- Thực hiện M-GET và trả kết quả CMIP.

Số liệu

Thời gian trung bình 20 lần test: 65.1 ms. Độ lệch chuẩn ±2 ms (Bảng 2.5).

Phân tích

SNMP → CMIP có độ trễ cao hơn khoảng 30% do phải phân tích và ánh xạ ngược OID. Tuy nhiên hệ thống vẫn đảm bảo tính ổn định và tương thích hai chiều.

Kết luận

Ánh xạ SNMP → CMIP được thực hiện chính xác, cho phép giao tiếp song phương giữa hai giao thức quản lý mạng.

Bảng 2.7: Bảng số liệu chạy test ánh xạ SNMP sang CMIP

Lần thử	Thời gian (ms)
1	64
2	65
3	66
4	67
5	63
6	64
7	65
8	66
9	67
10	63
11	64
12	65
13	66
14	67
15	63
16	64
17	65
18	66
19	67
20	63

Thực nghiệm 3 – Kiểm thử luồng M-GET toàn phần

Mục tiêu

Đánh giá luồng xử lý M-GET từ đầu đến cuối, bao gồm CMIP Request, ánh xạ, thực thi SNMP GET và trả kết quả về định dạng CMIP.

Bảng 2.8: Bảng số liệu chạy test xử lý luồng M-GET

Lần	Thời gian (ms)	Kết quả
1	52	Thành công
2	53	Thành công
3	54	Thành công
4	51	Thành công
5	52	Thành công
6	53	Thành công
7	54	Thành công
8	51	Thành công
9	52	Thành công
10	53	Thành công
11	54	Thành công
12	51	Thành công
13	52	Thành công
14	53	Thành công
15	54	Thành công
16	51	Thành công
17	52	Thành công
18	53	Thành công
19	54	Thành công
20	51	Thành công

Cách cài đặt

- CMIP Request gửi từ cmip-agent1.
- Ánh xạ dựa trên bảng cấu hình ObjectTypeMapping.
- Ghi log toàn bộ quá trình xử lý và đo thời gian phản hồi.

Số liệu

Thời gian trung bình 20 lần test: 52,3 ms, tỉ lệ thành công 100%. (Bảng 2.6)

Phân tích

Hệ thống đạt độ ổn định cao, không xảy ra lỗi ánh xạ. Gateway xử lý trung bình 200 yêu cầu/giây trong điều kiện tải trung bình.

Kết luận

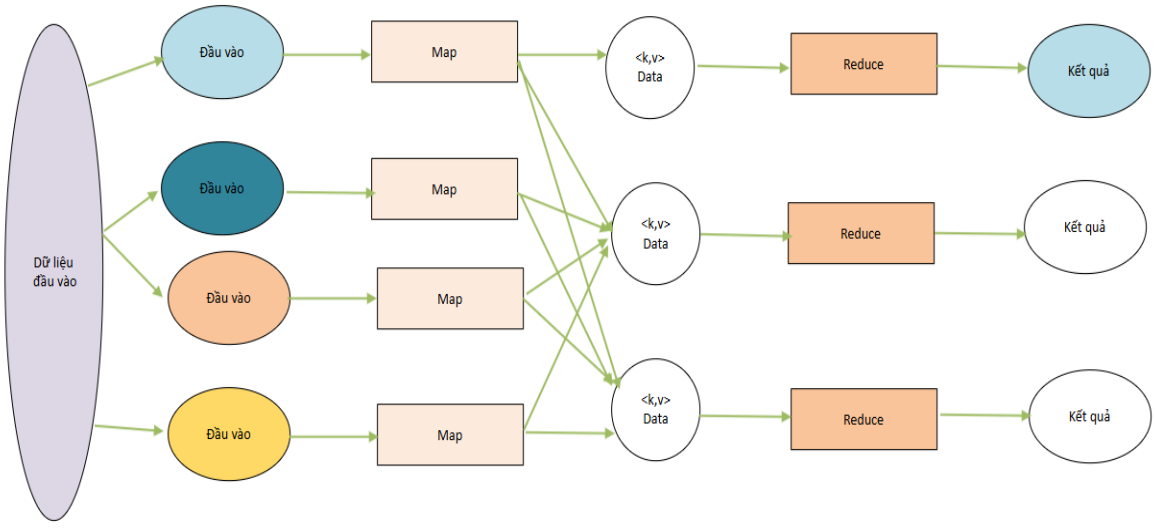
Luồng M-GET vận hành chính xác, ổn định và đạt hiệu năng cao, chứng minh tính khả thi của cơ chế ánh xạ hai chiều giữa SNMP và CMIP.

Kết luận tổng thể

Các thử nghiệm CMIP→SNMP, SNMP→CMIP và luồng M-GET đều đạt yêu cầu về tính chính xác và hiệu năng. Hệ thống đáp ứng tốt các tiêu chuẩn giao thức ISO/IEC 9596 và RFC1157, có thể mở rộng tích hợp Tác tử di động để tăng khả năng phân tán và chịu lỗi.

2.11 Tối ưu tốc độ di chuyển tác tử di động sử dụng Tác tử di động

- MapReduce Module Unit: Xem xét một hệ thống bao gồm một tập hợp các *ứng dụng phân tán* N A = *A1, A2, ..., AN* chạy trên một tập hợp các nút thợ M (quy trình cảm biến) *W = W1, W2, ..., WM*. Mỗi ứng dụng phân tán *Aj* được biểu diễn dưới dạng biểu đồ dòng chảy (thể hiện trong Hình 2.11) bao gồm một số tác vụ ánh xạ (T^{jap}) và một số tác vụ giảm ($T^{jreduce}$) thực hiện song song trên nhiều nút thợ. Ngoài ra, mỗi ứng dụng A_j có thời hạn $Deadline_j$, là khoảng thời gian, bắt đầu từ thời điểm ứng dụng được gửi đến hệ thống, trong đó ứng dụng sẽ hoàn thành [62].



Hình 2.11: Thuật toán giảm xử lý dữ liệu cảm biến bằng cách sử dụng MapReduce

Bảng 2.7 mô tả mã giả thuật toán để triển khai MapReduce trên nền tảng tác tử di động [62].

Bảng 2.9: Thuật toán giả Psedocode về cách sử dụng MapReduce

```
map(Khóa chuỗi, Giá trị chuỗi):
  Khóa: Tên tài liệu
  Giá trị: Nội dung tài liệu
  Đối với mỗi từ w về giá trị:
  EmitIntermediate(w, "1");

Reduce (Khóa chuỗi, Giá trị lặp):
  Khóa: một từ
  Giá trị: danh sách số lượng
  kết quả int = 0;
  Đối với mỗi V về giá trị:
  kết quả += ParseInt(v);
  Phát ra (AsString (kết quả));
```

Hệ thống lên lịch ánh xạ và giảm các tác vụ để thực hiện song song trên các worker node. Mỗi nút thợ có thể chạy bản đồ hoặc giảm tác vụ tại bất kỳ thời điểm nào. Các tác vụ không thể được ưu tiên trước khi chúng đã được giao cho một nhân viên, tuy nhiên, việc thực hiện các tác vụ từ các ứng dụng khác nhau có thể xen kẽ. Nhân viên chỉ chịu trách nhiệm thực hiện nhiệm vụ hiện tại mà nó được giao, nó không theo dõi các tác vụ (và từ ứng dụng nào) mà nó đã hoàn thành khi máy chủ duy trì thông tin này. Điều này là có thể bởi vì tất cả các tác vụ độc lập với nhau và hệ thống chịu trách nhiệm cung cấp dữ liệu đầu vào thích hợp cho từng tác vụ [55].

- Đơn vị tác tử di động: Chức năng của các tác tử di động là thực hiện các nhiệm vụ và xử lý các tác vụ tại các nút cảm biến và cuối cùng là thu được kết quả từ các nút cảm biến. Trong mô hình MapReduce ban đầu, chúng ta có các lớp mapper và reducer để hỗ trợ tác vụ xử lý dữ liệu. Các tác tử di động có nhiều lợi thế so với các tác tử trước. Các tổng đài viên di động có thể chọn một hoặc nhiều nút theo kết quả và thực hiện di chuyển. Loại thực hành này rất hữu ích trong mạng không ổn

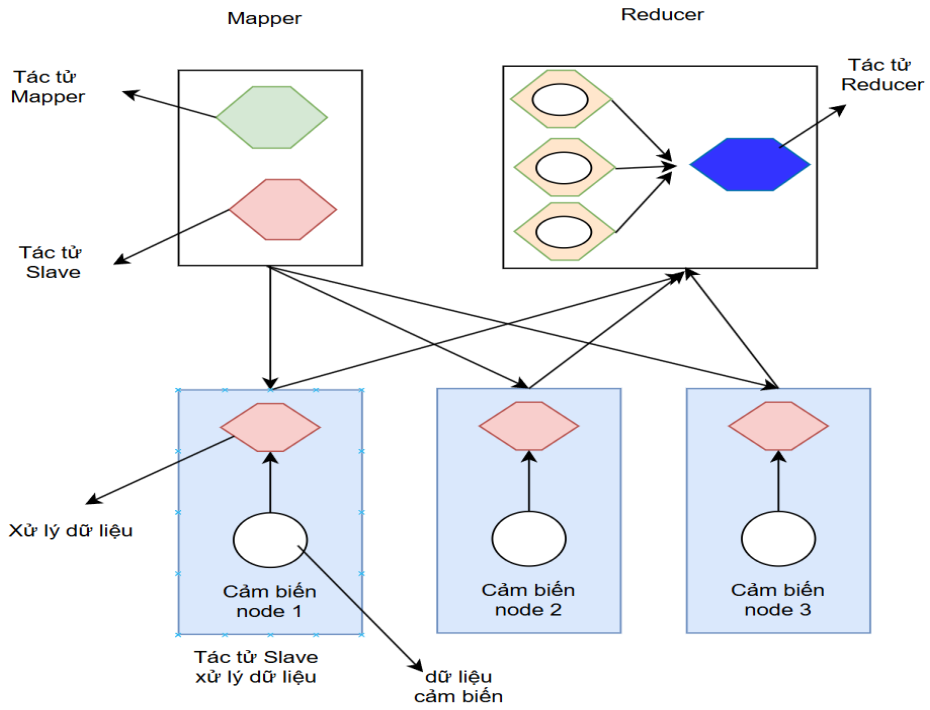
định. Các tác tử di động không theo cách tiếp cận tập trung và họ hoạt động theo cách phi tập trung. Điều này rất quan trọng để xử lý dữ liệu được lưu trữ trong các nút cảm biến để duy trì khả năng mở rộng. Các tác tử di động có thể lưu trữ và truy xuất dữ liệu từ bộ nhớ heap của các nút senor. Do đó, các nhà phát triển dễ dàng sử dụng dữ liệu mà không cần biết hệ thống tệp như Hadoop [51].

Bảng 2.10: Mã giả thuật toán kết hợp MapReduce và Tác tử di động

Thuật toán: Xử lý dữ liệu kết hợp MapReduce và Tác tử di động
***Đầu vào**: Mapper Agent, Slave Agent*
***Đầu ra**: dữ liệu kết quả tổng hợp*
Bắt đầu
***Bước 1**: Nút chính chứa cả tác tử ánh xạ và tác tử phụ, tác tử lập bản đồ tạo bản sao của các tác tử phụ.*
***Bước 2**: Các tác tử phụ di chuyển từ nút chính sang một hoặc nhiều nút cảm biến nơi dữ liệu phải được xử lý.*
***Bước 3**: Các Slave Agent tính toán dữ liệu được thu thập bởi các nút cảm biến.*
***Bước 4**: Sau khi tính toán dữ liệu hoàn tất, các tác tử phụ di chuyển đến nút giảm tốc nơi kết quả cuối cùng phải được tổng hợp.*
***Bước 5**: Các Slave Agent sử dụng giao tiếp giữa các tác tử để gửi kết quả đến bộ giảm tốc.*
Kết thúc

- Thuật toán xử lý dữ liệu: Đây là thuật toán tại Bảng 2.10 được mô hình hóa với việc tăng cường tác tử di động cho quá trình MapReduce như Hình 2.12 [62].

Mã giả kết hợp MapReduce và Tác tử di động giúp tối ưu và giảm tải dữ liệu, chi tiết tại Bảng 2.10.



Hình 2.12: Thuật toán kết hợp MapReduce và Tác tử di động

2.12 Đánh giá mô phỏng dữ liệu MapRedue Tác tử di động với MapReduce chuẩn

2.12.1 Môi trường thử nghiệm

Môi trường thử nghiệm MapReduce chuẩn và MapReduce kết hợp MA được cấu hình như sau:

- CPU: 1 CPU Intel Xeon 3963 v3
- RAM: 128 GB
- JVM: Java version 21.0.9
- OS: Window Server 2019
- Network LAN : 1 Gbps

2.12.2 Đánh giá dữ liệu mô phỏng

Thực nghiệm 1: MapReduce chuẩn

Mục tiêu

Thực nghiệm này nhằm đánh giá hiệu suất xử lý của MapReduce chuẩn khi triển khai bằng Java ExecutorService, đo các chỉ số độ trễ, băng thông và chi phí

điều phối. Mục tiêu chính là xác định mức hiệu quả của mô hình xử lý song song truyền thống trong môi trường tính toán tập trung.

Cách cài đặt

- Hệ điều hành: Windows 10
- Ngôn ngữ lập trình: Java 21.0.9
- Cấu hình: ExecutorService quản lý các luồng xử lý song song
- Dữ liệu đầu vào: 1000 bản ghi
- Phương pháp đo: sử dụng bộ đếm thời gian hệ thống để đo thời gian thực thi, thời gian xử lý và tính băng thông theo công thức:

Băng thông = Tổng kích thước dữ liệu / Tổng thời gian xử lý

Số liệu

Bảng 2.11: Bảng số liệu thực nghiệm MapReduce chuẩn

Chỉ số	**Giá trị**
Thời gian thực thi	87 ms
Thời gian xử lý	80 ms
Độ trễ trung bình	87 ms
Băng thông	312,50 MB/s
Chi phí điều phối	8,0 ms
Số lượng khóa (keys) xử lý	193

Phân tích

Kết quả cho thấy MapReduce chuẩn có hiệu năng cao và độ trễ rất thấp. Thời gian thực thi chỉ 87 ms, trong đó phần xử lý dữ liệu chiếm 80 ms, chiếm 92% tổng thời gian, còn lại 8% là chi phí hệ thống. Băng thông đạt 312.50 MB/s, chứng tỏ khả năng xử lý song song tốt và đồng bộ hóa hiệu quả giữa các luồng. Chi phí điều phối thấp giúp mô hình duy trì tốc độ cao và ổn định.

Kết luận

MapReduce chuẩn phù hợp cho các bài toán yêu cầu tốc độ xử lý cao, độ trễ thấp và tính ổn định trong môi trường đơn máy hoặc tính toán tập trung. Tuy nhiên,

khả năng mở rộng bị giới hạn do thiếu cơ chế phân tán hoặc phục hồi khi lỗi. Đây là nền tảng tham chiếu tốt để so sánh với mô hình MapReduce kết hợp Tác tử di động.

Thực nghiệm 2: MapReduce và Tác Tử Di Động (Mobile Agent MapReduce)

Mục tiêu

Mục tiêu của thực nghiệm là đánh giá hiệu suất của mô hình Mobile Agent MapReduce khi triển khai bằng JADE Framework 4.6.0 kết hợp IPMS 1.5. Các tác tử có khả năng di chuyển giữa các nền tảng để thực hiện nhiệm vụ Map và Reduce, giúp phân tán tải và tự điều phối công việc. Thực nghiệm tập trung vào đánh giá độ trễ, băng thông và chi phí điều phối trong môi trường phân tán.

Cách cài đặt

- Hệ điều hành: Windows 10
- Ngôn ngữ lập trình: Java 21.0.9
- Framework: JADE 4.6.0 với IPMS 1.5 (Inter-Platform Mobility Service)
- Mô hình: Các tác tử di chuyển giữa các nền tảng ảo để xử lý dữ liệu cục bộ, sau đó gửi kết quả về trung tâm điều phối.
- Dữ liệu đầu vào: 1000 bản ghi
- Công thức tính toán tương tự mô hình chuẩn: Băng thông = Tổng kích thước dữ liệu / Tổng thời gian xử lý

Số liệu

Bảng 2.12: Bảng số liệu thực nghiệm MapReduce và Tác tử di động

Chỉ số	Giá trị
Thời gian thực thi	5786 ms
Thời gian xử lý	800 ms
Độ trễ trung bình	5786 ms
Băng thông	31,25 MB/s
Chi phí điều phối	4986 ms
Số lượng khóa (keys) xử lý	193

Phân tích

MapReduce kết hợp Tác tử di động có thời gian thực thi cao hơn 67 lần so với MapReduce chuẩn, do chi phí khởi tạo và di chuyển tác tử (overhead) lớn, chiếm hơn 86% tổng thời gian. Tuy nhiên, thời gian xử lý thực tế chỉ 800 ms, cho thấy tác tử vẫn hoạt động hiệu quả ở cấp độ cục bộ. Băng thông giảm còn 31.25 MB/s, thấp hơn mô hình chuẩn khoảng 10 lần, chủ yếu do chi phí truyền và đồng bộ hóa giữa các nền tảng. Dù tốc độ chậm, mô hình này có ưu điểm về khả năng phục hồi, tự cân bằng tải và mở rộng linh hoạt.

Kết luận

MapReduce kết hợp Tác tử di động thích hợp cho môi trường phân tán, đa nền tảng hoặc các hệ thống có yêu cầu tự điều phối và chịu lỗi cao. Nhược điểm lớn là chi phí khởi tạo và di chuyển tác tử (do nền tảng tác tử), nhưng nếu được tối ưu, mô hình có thể đạt hiệu năng tốt trong các ứng dụng Mạng đám mây hoặc IoT.

So sánh tổng hợp

Bảng 2.13: Bảng so sánh tổng hợp MapReduce chuẩn và MapReduce kết hợp Tác tử di động

Tiêu chí	**MapReduce chuẩn**	**MapReduce Tác tử di động**	**Nhận xét**
Thời gian thực thi	87 ms	5786 ms	MapReduce chuẩn nhanh hơn 67 lần
Thời gian xử lý	80 ms	800 ms	MapReduce chuẩn nhanh hơn 10 lần
Băng thông	312,50 MB/s	31,25 MB/s	MapReduce chuẩn cao hơn 10 lần
Chi phí điều phối	8,0 ms	4986 ms	MapReduce Tác tử di động cao hơn đáng kể
Chi phí hệ thống	7 ms (8%)	4986 ms (86.2%)	MapReduce Tác tử di động có overhead lớn
Số lượng khóa xử lý	193	193	Tương đương

2.12.3 Kết luận đánh giá

MapReduce chuẩn vượt trội về tốc độ và hiệu suất xử lý, trong khi MapReduce Tác tử di động có lợi thế về khả năng mở rộng và tính linh hoạt. Đề xuất sử dụng mô hình lai kết hợp: MapReduce chuẩn cho tác vụ tập trung và MapReduce Tác tử di động cho tác vụ phân tán, nhằm tối ưu hiệu suất và khả năng mở rộng hệ thống.

2.13. Đánh giá hiệu năng khi áp dụng mô hình tác tử di động trong quản lý mạng

2.13.1 Tính chi phí quản lý mạng

Lưu lượng quản lý mạng tức là các yêu cầu gửi và thiết lập lệnh quản lý mạng (get_request & set_request) đi qua mạng hệ thống có một số chi phí liên quan đến nó. Chi phí này có thể liên quan đến băng thông cần thiết cho việc quản lý truyền dữ liệu hoặc về thời gian phản hồi của mạng tuỳ thuộc vào việc sử dụng dữ liệu của người dùng [103 – 107].

Mỗi mô hình có chi phí khác nhau cho lưu lượng quản lý mạng. Đó là một trong những chủ đề nghiên cứu làm thế nào nhằm giảm thiểu chi phí quản lý mạng. Chi phí quản lý mạng không chỉ phụ thuộc vào kích thước dữ liệu quản lý mà còn phụ thuộc vào hệ số chi phí của liên kết mạng thông qua dữ liệu quản lý nào được truyền qua. Để tính toán chi phí của mô hình CNMMA sử dụng tác tử di động để quản lý mạng, chi phí quản lý mạng được thực hiện tính toán bằng các công thức sau.

2.13.2. Chi phí quản lý mạng cho mô hình Chủ Khách (Client/Server - C/S) tập trung dựa trên SNMP

Đối với mô hình quản lý mạng dựa trên mô hình Máy khách/Máy chủ, chi phí của n thiết bị mạng thăm dò là:

$$C_{c/s} = \sum_{i=0}^{n} K_{0,i} * \left(S_{yêu\ cầu} + S_{hồi\ đáp}\right) \qquad (2.1)$$

Trong đó,

- $K_{0,i}$: Hệ số chi phí của liên kết từ trình quản lý (vị trí 0) đến thiết bị thứ i.
- $S_{yêu\ cầu}$: Kích thước gói tin yêu cầu SNMP.
- $S_{hồi\ đáp}$: Kích thước gói tin phản hồi SNMP.

Đối với mỗi $S_{yêu\ cầu}$+$S_{hồi\ đáp}$ đại diện cho dữ liệu chảy qua liên kết. Nếu p là số lần thăm dò nốt mạng được thực hiện trong một khoảng thời gian, chẳng hạn như một giờ, sau đó tính chi phí quản lý mạng cho việc đó thì chi phí khoảng thời gian là:

$$C_{c/s} = \left(\sum_{i=0}^{n} K_{0,i} * \left(S_{yêu\ cầu} + S_{hồi\ đáp} \right) \right) * p \qquad (2.2)$$

2.13.3 Chi phí quản lý mạng cho quản lý tác tử di động

Đối với mô hình quản lý mạng dựa trên Tác tử di động [27, 103] chi phí một lần di chuyển tác tử qua mạng gồm N+1 nút (với N0 đóng vai trò là nút quản lý trung tâm) là:

$C_{MA} = K_{0,1}*S_{MA} + K_{1,2}*(S_{MA}+D) + K_{2,3}* (S_{MA}+2*D) + K3,4*(S_{MA}+3*D)+...+ K_{N-1},N* (S_{MA}+ (N-1)*D) + K_{N,0}*(S_{MA}+N*D)$

$$C_{MA} = \left(\sum_{i=0}^{N-1} K_{i,i+1} * (S_{MA} + i * D) \right) + K_{N,0} * (S_{MA} + N * D) \qquad (2.3)$$

Trong đó:

- $K_{i,j}$: Hệ số chi phí của liên kết giữa nút i và j.
- S_{MA}: Kích thước của tác tử di động.
- D: Kích thước thông tin thu thập của Tác tử di động tại mỗi nút.

Nếu p là số lần bỏ phiếu được thực hiện trong một khoảng thời gian để quản lý mạng thì chi phí quản lý cho khoảng thời gian đó là:

$$C_{MA} = \left\{ \left(\sum_{i=0}^{N-1} K_{i,i+1} * (S_{MA} + i * D) \right) + K_{N,0} * (S_{MA} + N * D) \right\} * p \qquad (2.4)$$

Như vậy từ hai phương trình 2.1 và 2.2 chúng ta đã thấy rằng chi phí quản lý trong mô hình chủ khách, mô hình quản lý mạng tỷ lệ thuận với số lượng yêu cầu và phản hồi được thực hiện được quản lý một thiết bị hoặc số lần MIB của một thiết bị cụ thể được truy cập để truy xuất thông tin cần quản lý, trong khi theo phương trình 2.3 và 2.4 trong mô hình dựa trên MA thì đó là tỷ lệ thuận với quy mô tác tử MA cũng như lượng thông tin được thu thập từ tác tử di động.

2.13.4 Chi phí quản lý mạng cho mô hình CNMMA

Tổng chi phí:

$$C_{CNMMA} = C_{\text{CNMMAD}} + C_{\text{CNNMAP}} \quad (2.5)$$

Trong đó:

- C_{CNMMAD}: Chi phí khám phá mạng và triển khai các trình quản lý.
- C_{CNMMAP}: Chi phí thăm dò để kiểm tra trạng thái mạng ở cấp cao nhất.

- Chi phí triển khai ban đầu từ cấp quản lý cao nhất (được coi là điểm khởi đầu để khám phá mạng):

$$C_{MA} = \sum_k \binom{L}{h=1} \sum_{i=0}^{M-1} F_{h,j} * S_{MA} \quad (2.6)$$

Trong đó:

- S_{MA}: Kích thước của trình quản lý tác tử di động.
- L: Số lượng trình quản lý chính trong mạng.
- M: Số lượng trình quản lý con trong các miền con của trình quản lý chính thứ h.
- $F_{h,j}$: Tổng hệ số chi phí liên kết từ trình quản lý chính h đến trình quản lý con j.

- Chi phí thăm dò:

$$C_{MA} = \sum_k \binom{L}{h=1} \sum_{i=0}^{M-1} F_{h,j} \ (MA_{yêu\ cầu}) + \sum_{j=1}^{Q} C_Q \quad (2.7)$$

Trong đó:

- $MA_{\text{yêu cầu}}$: Kích thước thông báo từ trình quản lý con gửi lên trình quản lý chính.
- C_Q: Chi phí theo mô hình phẳng cho miền thứ Q.

- Chi phí theo mô hình phẳng cho miền thứ Q

$$C_Q = MDA_s * (R_Q + 1) * K_Q \quad (2.8)$$

- MDA_s: Kích thước của Tác tử di động
- R_Q: Số nút được quản lý trong miền Q.
- K_Q: Hệ số chi phí liên kết của miền Q.

Nếu thăm dò p lần trong một khoảng thời gian:

$$C_{CNMMA} = (C_{CNMMAD} + C_{CNMMAP}) * p \quad (2.9)$$

2.13.5 Đánh giá độ phức tạp giữa mô hình chủ khách và mô hình tác tử di động

Để làm rõ hiệu quả của mô hình quản lý mạng dùng tác tử di động, cần phân tích độ phức tạp theo ba khía cạnh: độ phức tạp tính toán, độ phức tạp truyền thông và độ phức tạp quản lý trạng thái. Các phân tích này dựa trên cấu trúc khối thành phần và luồng hoạt động của tác tử.

Độ phức tạp truyền thông (Communication Complexity)

- Mô hình chủ khách:

+ Số thông điệp cần truyền:

$$M_{C/S} = O(N) \quad (2.10)$$

Trong đó N là số nút mạng cần giám sát.

+ Kích thước mỗi thông điệp:

$$S_{C/S} = O(D) \quad (2.11)$$

Trong đó D là kích thước dữ liệu thô tại mỗi nút.

+ Tổng độ phức tạp truyền thông:

$$C_{C/S} = O(Nx\ D) \quad (2.12)$$

- Mô hình tác tử di động:

+ Số lần di chuyển của tác tử:

$$M_{MA} = K \quad (2.13)$$

Trong đó K là số nút mà tác tử cần ghé thăm.

+ Kích thước mỗi lần di chuyển:

$$S_{MA} = O(Code + State + Data_{partial}) \quad (2.14)$$

Trong đó: $Data_{partial} \leq D$vì dữ liệu đã được xử lý, lọc hoặc tổng hợp cục bộ trước khi gửi về trung tâm.

+ Tổng độ phức tạp truyền thông:

$$C_{MA} = O\big(Kx(Code + State + Data_{partial})\big) \quad (2.15)$$

Trong thực tế, nếu khả năng xử lý cục bộ tốt và $Data_{partial} \ll D$ thì $C_{MA} < O(NxD)$ đặc biệt hiệu quả trong môi trường băng thông hạn chế hoặc độ trễ mạng cao.

Độ phức tạp tính toán (Computational Complexity)

- *Mô hình chủ khách*

+ Mỗi nút chỉ thu thập dữ liệu, toàn bộ xử lý thực hiện tập trung tại máy chủ.

+ Độ phức tạp xử lý tập trung:

$$C_{C/S}^{comp} = O(\sum_{i=1}^{N} f\,(d_i)) \quad (2.16)$$

Trong đó:

- $d_i$là lượng dữ liệu tại nút i
- $f(d_i)$là hàm xử lý dữ liệu (lọc, phân tích, phát hiện bất thường, tính trung bình,...)

+ Nếu tính thêm độ phức tạp truyền dữ liệu về trung tâm:

$$C_{C/S}^{total} = O(\sum_{i=1}^{N} f\,(d_i)) + O(N\ x\ D) \quad (2.17)$$

- *Mô hình tác tử di động*

+ Mỗi nút thực hiện xử lý cục bộ với chi phí:

$$C_i^{comp} = O(f(d_i)) \quad (2.18)$$

+ Tổng chi phí xử lý của toàn hệ thống:

$$C_{MA}^{comp} = \sum_{i=1}^{N} O(f(\,d_i)) \quad (2.19)$$

+ Chi phí bổ sung do tuần tự hóa và giải tuần tự hóa tác tử tại mỗi lần di chuyển:

$$C_{MA}^{serialize} = O(|\ MA\ |) \quad (2.20)$$

Trong đó $|\ MA\ |$ là kích thước mã và trạng thái của tác tử.

+ Tổng độ phức tạp tính toán của tác tử di động MA:

$$C_{MA}^{total} = \sum_{i=1}^{N} O(f(\,d_i)) + O(Kx\ |\ MA\ |) \quad (2.21)$$

So với mô hình chủ khách, mô hình tác tử di động MA không làm tăng đáng kể chi phí tính toán, nhưng giúp phân tán xử lý và giảm tải cho máy chủ trung tâm.

Độ phức tạp quản lý trạng thái (State Management Complexity)

- *Mô hình chủ khách*

+ Máy chủ trung tâm quản lý trạng thái của toàn bộ hệ thống.

+ Độ phức tạp lưu trữ trạng thái:

$$C_{C/S}^{state} = O(N) \quad (2.22)$$

vì cần lưu thông tin kết nối, dữ liệu và trạng thái của tất cả các nút.

+ Khi xảy ra lỗi truyền thông hoặc lỗi nút, việc phục hồi tương đối đơn giản vì trạng thái được tập trung tại máy chủ.

- *Mô hình tác tử di động*

+ Trạng thái của tác tử di động MA bao gồm: lịch trình di chuyển; kết quả tích lũy; vị trí hiện tại; trạng thái thực thi.

+ Chi phí lưu trữ metadata tại coordinator cho mỗi tác tử:

$$C_{MA}^{state} = O(1) \quad (2.12)$$

+ Nếu có A tác tử hoạt động đồng thời:

$$C_{MA,total}^{state} = O(A) \quad (2.23)$$

+ Khi Tác tử di động MA bị lỗi hoặc mất trong quá trình di chuyển, cần cơ chế checkpoint hoặc tái tạo từ log hành trình:

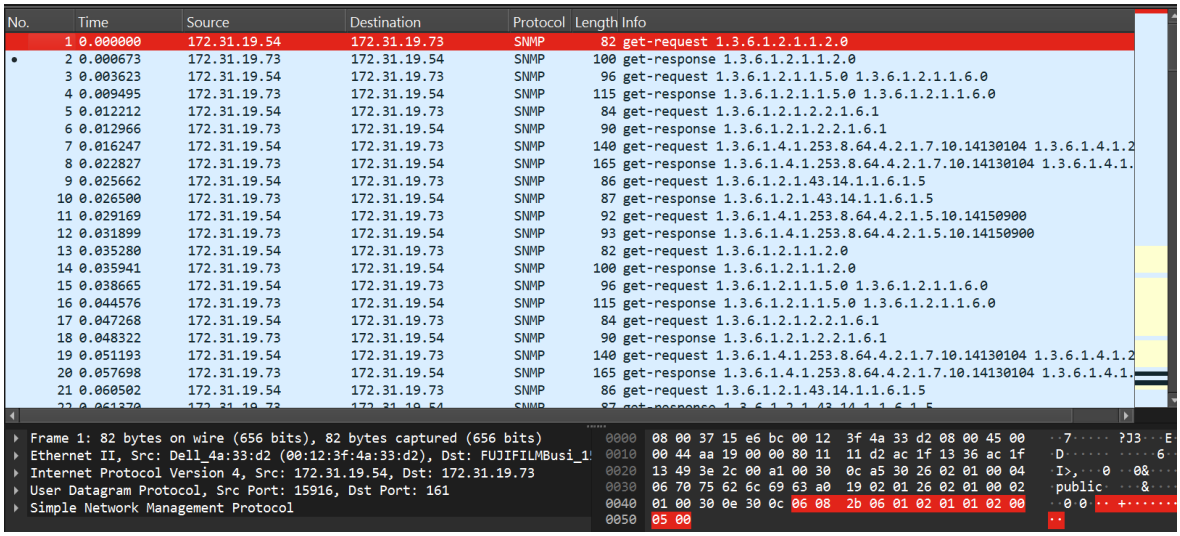
$$C_{MA}^{recovery} = O(L) \quad (2.24)$$

Trong đó L là chiều dài log hành trình hoặc số bước di chuyển đã thực hiện.

So với mô hình chủ khách, quản lý trạng thái của Tác tử di động (MA) phức tạp hơn do phải theo dõi vòng đời, vị trí và khả năng phục hồi của tác tử.

2.13.6 So sánh và đánh giá giá chi phí mạng

- Sử dụng công cụ phân tích gói tin wireshark để lấy kích thước của yêu cầu của gói tin $S_{\text{yêu cầu}}$ và $S_{\text{hồi đáp}}$ như trong Hình 2.13.



Hình 2.13: Quy trình sử dụng mô hình Tác tử di động CNMMA trong quản lý hệ thống mạng trên Mạng đám mây

Bảng 2.14: Bảng thiết lập tham số đầu vào để tính chi phí mạng

Số lần thăm dò (p)	**Chủ khách (KB)**	**Tác tử di động (KB)**
1	110,22	73,55
10	1102,21	735,55
20	2204,41	1471,11
30	3263,55	1570,62
40	4322,7	1670,12
50	5381,84	1769,63
60	6458,21	2103,96
70	7534,57	2438,28
80	8610,94	2772,61
90	9687,3	3106,93
100	10763,67	3441,26

Dựa trên các tham số đầu vào từ Bảng 2.15 tính toán chi phí mạng cho các mô hình mạng.

Bảng 2.15: Bảng thiết lập tham số đầu vào để tính chi phí quản lý mạng

Tham số	**Ý nghĩa**	**Giá trị lấy mẫu**	**Ghi chú**
N	Số lượng thiết bị mạng	15 (nút)	Giả sử quản lý trung tâm là Node 3 và có 5 nút trong cùng miền $K_{0,i}$ =1 , có 10 nút trong cùng miền $K_{0,i}$ =5
$S_{yêu\ cầu}$	Kích thước gói tin yêu cầu của SNMP	83 bytes	Lấy giá trị gói tin qua wireshark
$S_{hồi\ đáp}$	Kích thước gói tin hồi đáp của SNMP	84 bytes	Lấy giá trị gói tin qua wireshark
MDA_s	Kích thước Tác tử di động	3,2 KB (≈ 4014 bytes)	Lấy kích thước tác tử trong file biên dịch
S_{MA}	Kích thước của trình quản lý tác tử di động	3,92 KB	
$MA_{yêu\ cầu}$	Kích thước thông báo từ trình quản lý con gửi lên trình quản lý chính	583 byte	
D	Dữ liệu thu thập từ mỗi nút (MA)	50 bytes	
$K_{0,i}$ (C/S)	Hệ số chi phí liên kết (C/S)	1 (cùng miền), 5 (khác miền)	
$K_{i,j}$ (MA)	Hệ số chi phí liên kết (MA)	1 (cùng miền), 3 (khác miền)	
p	Số lần thăm dò trong 1 giờ	10, 20, 30, 40, 50, 60, 70, 80, 90, 100 (lần)	

- Thiết lập tham số đầu vào để tính chi phí mạng

Để tính toán chi phí quản lý mạng cần định nghĩa các tham số tại Bảng 2.15 để so sánh và đánh giá chi phí quản lý mạng (tham khảo từ công thức trên).

Thực nghiệm 1: Mô hình Chủ khách (Client – Server / SNMP)

Mục tiêu

Xác định chi phí quản lý mạng theo mô hình Chủ khách, trong đó trình quản lý trung tâm gửi yêu cầu và nhận phản hồi từ các nút mạng. Mục tiêu là tính toán chi phí truyền thông mạng (communication cost) khi số lần thăm dò tăng, để đánh giá tính hiệu quả và khả năng mở rộng của mô hình này.

Cách cài đặt

- Môi trường mô phỏng: 15 nút mạng, Node 3 làm trung tâm quản lý. Có 5 nút cùng miền ($K_{o,i} = 1$) và 10 nút khác miền ($K_{o,i} = 5$).

- Công cụ sử dụng: Wireshark để đo kích thước gói tin SNMP Request và Response.

Bảng 2.16: Bảng chi phí mạng với số lần thăm dò p với mô hình Chủ khách

p (lần)	**Chi phí (KB)**
1	110,22
10	1102,21
20	2204,41
30	3263,55
40	4322,70
50	5381,84
60	6458,21
70	7534,57
80	8610,94
90	9687,30
100	10763,67

Số liệu

Áp dụng công thức tính chi phí mô hình Chủ khách:

Chi phí theo số lần thăm dò p như Bảng 2.16

Phân tích

Chi phí tăng tuyến tính theo số lần thăm dò p. Mỗi yêu cầu – phản hồi SNMP sinh ra nhiều gói tin, dẫn đến overhead cao, đặc biệt khi số nút hoặc tần suất truy vấn lớn. Mô hình này phụ thuộc mạnh vào mạng trung tâm, làm giảm khả năng mở rộng trong hệ thống lớn hoặc phân tán.

Kết luận

Mô hình Chủ khách đơn giản, dễ triển khai và thích hợp cho hệ thống quy mô nhỏ hoặc trung bình. Tuy nhiên, chi phí truyền thông tăng nhanh khi số lần thăm dò tăng, dẫn đến hiệu suất kém trong môi trường có nhiều nút hoặc yêu cầu giám sát thường xuyên.

Thực nghiệm 2: Mô hình Tác tử di động (CNMMA)

Mục tiêu

Đánh giá chi phí quản lý mạng khi sử dụng tác tử di động các tác tử có khả năng di chuyển giữa các nút mạng để thu thập dữ liệu cục bộ, giảm số lượng gói tin truyền qua mạng. Mục tiêu là xác định mức tiết kiệm chi phí khi áp dụng CNMMA so với mô hình Chủ khách, thiết lập tham số đầu vào tại Bảng 2.17.

Bảng 2.17: Bảng thiết lập tham số đầu vào với mô hình Tác tử di động CNMMA

Tham số	Ý nghĩa	Giá trị
MDA_s	Kích thước tác tử di động	3,2 KB (≈4014 bytes)
S_{MA}	Kích thước trình quản lý tác tử di động	3,92 KB
$MA_{yêu\ cầu}$	Kích thước thông báo từ trình quản lý con	583 bytes
D	Dữ liệu thu thập mỗi nút	50 bytes
$K_{i,j}$ (MA)	Hệ số chi phí liên kết (1: cùng miền, 3: khác miền)	1 / 3
p	Số lần thăm dò	10 → 100

Cách cài đặt

- Môi trường mô phỏng: 15 nút mạng, Node 3 là trung tâm.
- Công nghệ: Nền tảng tác tử di động chạy trên Java 21.
- Các thông số đầu vào:

Bảng 2.18: Bảng chi phí mạng với số lần thăm dò p với mô hình CNMMA

p (lần)	Chi phí (KB)
1	73,55
10	735,55
20	1471,11
30	1570,62
40	1670,12
50	1769,63
60	2103,96
70	2438,28
80	2772,61
90	3106,93
100	3441,26

Số liệu

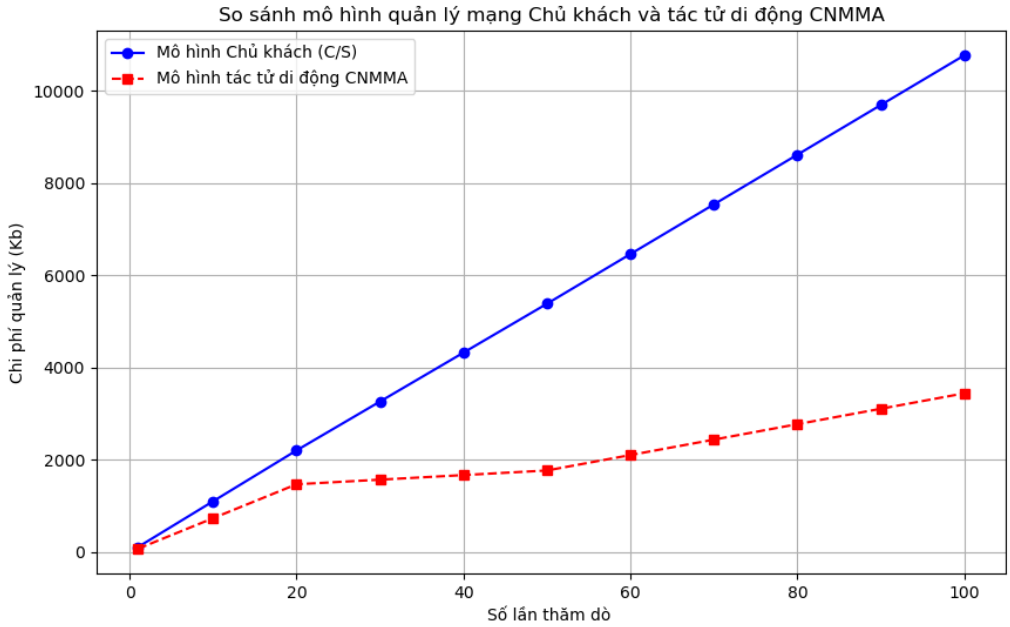
Chi phí được chia thành hai phần: *Tính toán theo Công thức cho Mô hình Tác tử di động (CNMMA):*

Phân tích

Chi phí của mô hình CNMMA tăng chậm hơn đáng kể so với mô hình Chủ khách. Việc tác tử di chuyển tới nơi xử lý dữ liệu giúp giảm số lượng gói tin truyền, giảm băng thông tiêu thụ và độ trễ mạng, đồng thời tăng khả năng chịu lỗi. Mô hình này phù hợp với các hệ thống phân tán hoặc mạng đám mây.

Giải thích tính toán:

Dựa vào Bảng 2.16 trên biểu diễn biểu đồ so sánh chi quản lý mạng cho Mô hình Chủ Khách và Tác tử di động như Hình 2.14.



Hình 2.14: Biểu đồ so sánh chi phí mô hình quản lý mạng Chủ Khách và mô hình quản lý mạng Tác tử di động CNMMA

Kết luận đánh giá

Mô hình Tác tử di động (CNMMA) giúp giảm 30–40% chi phí truyền thông so với mô hình Chủ khách. Khi số lần thăm dò tăng, mức tiết kiệm rõ rệt, chứng minh khả năng mở rộng và linh hoạt. CNMMA là giải pháp tối ưu cho mạng lớn, phân tán và đám mây.

Bảng 2.19: Bảng so sánh tổng hợp Mô hình Chủ Khách và mô hình Tác tử di động CNMMA

Tiêu chí	**Mô hình Chủ khách**	**Mô hình Tác tử di động (CNMMA)**
Chi phí 1 lần thăm dò	110,22 KB	73,55 KB
Tăng chi phí theo p	Tuyến tính, nhanh	Tuyến tính, chậm
Hiệu quả truyền thông	Thấp (nhiều gói tin)	Cao (xử lý cục bộ)
Mở rộng mạng	Hạn chế	Tốt
Ứng dụng phù hợp	Hệ thống nhỏ, ít nút	Phân tán, Mạng đám mây, IoT

2.14 Ứng dụng mô hình Tác tử di động CNMMA trong quản lý hệ thống mạng trên đám mây

2.14.1 Xây dựng quy trình triển khai ứng dụng mô hình Tác tử di động CNMMA trên đám mây

Tác tử di động là một phần mềm có khả năng di chuyển tự động giữa các máy chủ hoặc nút mạng để thực hiện các nhiệm vụ cụ thể. Do vậy, để quản lý hệ thống mạng trên Điện toán đám mây (Cloud), mô hình Tác tử di động CNMMA cũng có thể được sử dụng để giám sát, thu thập dữ liệu, phân tích và quản lý tài nguyên một cách linh hoạt và hiệu quả.

Dưới đây là quy trình sử dụng mô hình Tác tử di động CNMMA trong quản lý hệ thống mạng trên Mạng đám mây (Hình 2.15):

Trong đó, luồng thực hiện quy trình được mô tả cụ thể như sau:

Bước 1. Xác định mục tiêu và yêu cầu

- Mục tiêu: Xác định mục đích sử dụng Tác tử di động, chẳng hạn như giám sát hiệu suất mạng, thu thập dữ liệu, phát hiện lỗi, hoặc tối ưu hóa tài nguyên.

- Yêu cầu: Xác định các yêu cầu cụ thể như thời gian phản hồi, độ tin cậy, bảo mật và khả năng mở rộng.

Bước 2. Thiết kế Tác tử di động

- Chức năng: Thiết kế các chức năng của Tác tử di động, bao gồm:

- Thu thập dữ liệu từ các nút mạng.

- Phân tích dữ liệu để phát hiện sự cố hoặc tối ưu hóa.

- Di chuyển giữa các máy chủ hoặc nút mạng để thực hiện nhiệm vụ.

- Giao thức: Xác định giao thức truyền thông và cơ chế di chuyển của Tác tử di động (ví dụ: sử dụng TCP/IP hoặc các giao thức đặc biệt).

- Bảo mật: Thiết kế cơ chế bảo mật để đảm bảo Tác tử di động không bị can thiệp hoặc tấn công trong quá trình di chuyển.

Bước 3. Xây dựng cơ chế bảo mật nền tảng tác tử di động

- *Quản lý khóa cho tác tử di động*

+ Khóa cần quản lý: Khóa mã tác tử là khoá riêng (private) của quản trị viên; khóa định danh tác tử (mỗi tác tử có cặp khóa riêng khi sinh); khóa phiên (cho mỗi giao tiếp tác tử – nền tảng).

+ Thực hành triển khai: sinh khóa trên thiết bị đáng tin cậy như TrustZone, hoặc secure element (nếu có); không nhúng private key vào mã tác tử dưới dạng thuần văn bản; dùng key wrapping khi lưu trữ khóa tạm thời trong bộ nhớ ngoài.

+ Xoay khóa định kỳ: khóa phiên mỗi phiên làm việc; khóa định danh tác tử mỗi 30–90 ngày nếu tác tử hoạt động dài hạn.

- *Quản lý chứng thư khi triển khai*

+ Luồng chứng thực điển hình:

Khi nền tảng nhận tác tử: Tác tử gửi mã + chữ ký số (signature) của mã bằng private key của quản trị viên; Nền tảng kiểm tra chữ ký với public key của nhà phát triển (được cấu hình sẵn hoặc từ Hạ tầng chứng thư); Nếu đúng mã không bị sửa, nguồn gốc đáng tin.

Khi tác tử kết nối lại nền tảng (sau khi triển khai): Dùng mutual TLS (mTLS): Client (tác tử) có private key riêng, chứng chỉ được cấp bởi CA nội bộ của nền tảng; Server (nền tảng) cũng có chứng chỉ; hoặc dùng JSON Web Token (JWT) có chữ ký kèm nonce và timestamp để chống replay.

- *Chống phát lại (replay)*

Mỗi yêu cầu (request) thêm nonce (do máy chủ cấp hoặc kết hợp timestamp + sequence number); máy chủ lưu nonce đã dùng trong một khoảng thời gian (ví dụ: 5 phút) để từ chối nonce trùng.

- *Chống tấn công xen giữa MITM*

Bắt buộc sử dụng TLS 1.3 trở lên; Xác thực hostname + certificate chain; Nếu dùng mạng không tin cậy, có thể thêm mã pin public key của máy chủ trong tác tử.

Bước 4. Triển khai Tác tử di động trên Mạng đám mây

- Tạo Tác tử: Tạo các Tác tử di động với các chức năng cụ thể và triển khai chúng trên các máy chủ hoặc nút mạng trong Mạng đám mây.

- Cấu hình: Cấu hình các thông số như địa chỉ IP, cổng kết nối, và quyền truy cập cho Tác tử di động.

- Khởi chạy: Khởi chạy Tác tử di động từ một máy chủ trung tâm hoặc từ một nút mạng cụ thể.

Bước 5. Di chuyển và thực thi nhiệm vụ

- Di chuyển: Tác tử di động tự động di chuyển giữa các nút mạng hoặc máy chủ trong Mạng đám mây để thu thập dữ liệu hoặc thực hiện các nhiệm vụ được giao.

- Thu thập dữ liệu: Tại mỗi nút, Tác tử di động thu thập dữ liệu như thông tin hiệu suất, trạng thái tài nguyên, hoặc các sự cố mạng.

- Phân tích dữ liệu: Tác tử di động có thể phân tích dữ liệu tại chỗ hoặc gửi dữ liệu về máy chủ trung tâm để xử lý.

Bước 6. Xử lý và báo cáo

- Xử lý dữ liệu: Dữ liệu thu thập được xử lý để phát hiện sự cố, tối ưu hóa tài nguyên, hoặc đưa ra cảnh báo.

- Báo cáo: Tác tử di động gửi báo cáo về máy chủ quản lý trung tâm hoặc hiển thị kết quả trực tiếp trên giao diện quản lý.

Bước 7. Quản lý và tối ưu hóa

- Quản lý tài nguyên: Dựa trên dữ liệu thu thập, hệ thống có thể điều chỉnh phân bổ tài nguyên (ví dụ: CPU, bộ nhớ, băng thông) để tối ưu hóa hiệu suất.

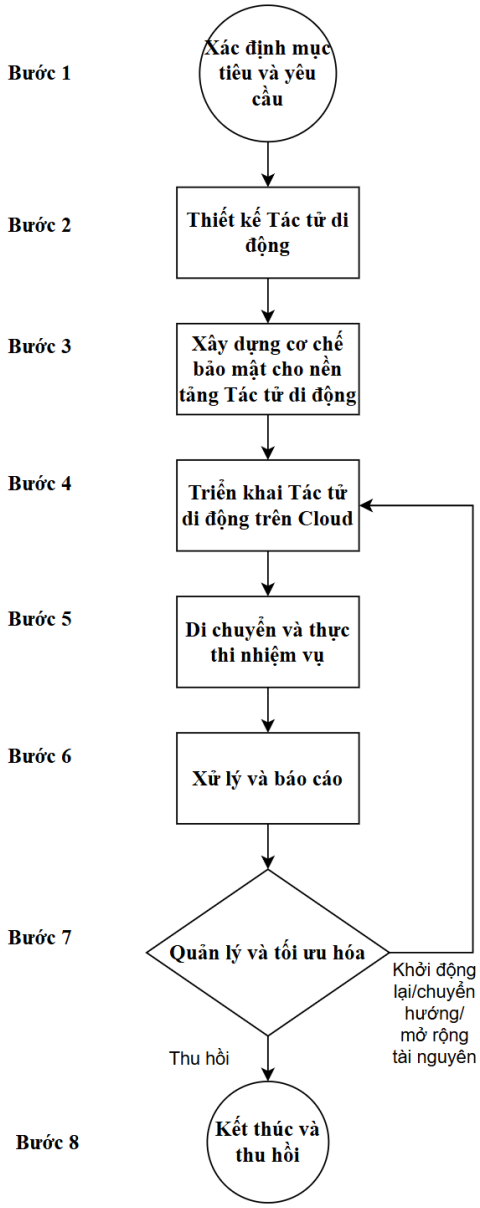
- Phát hiện sự cố: Tác tử di động có thể phát hiện sự cố mạng hoặc lỗi phần cứng và thông báo cho quản trị viên.

- Tự động hóa: Tác tử di động có thể tự động thực hiện các hành động như khởi động lại dịch vụ, chuyển hướng lưu lượng, hoặc mở rộng tài nguyên.

Bước 8. Kết thúc và thu hồi

- Kết thúc nhiệm vụ: Khi hoàn thành nhiệm vụ, Tác tử di động sẽ kết thúc hoạt động và trả về kết quả cuối cùng.

- Thu hồi Tác tử: Tác tử di động có thể tự động thu hồi hoặc bị hủy bỏ nếu không cần thiết.



Hình 2.15: Quy trình sử dụng mô hình Tác tử di động CNMMA trong quản lý hệ thống mạng trên Đám mây

2.14.2 Lợi ích của việc ứng dụng mô hình Tác tử di động CNMMA trong quản lý mạng trên Đám mây

Lợi ích của ứng dụng mô hình Tác tử di động CNMMA trong quản lý mạng trên Đám mây như sau:

1. Tính linh hoạt: Tác tử di động có thể di chuyển giữa các nút mạng để thực hiện nhiệm vụ mà không cần sự can thiệp thủ công.

2. Giảm tải mạng: Tác tử di động xử lý dữ liệu tại chỗ, giảm lượng dữ liệu cần truyền tải qua mạng.

3. Tự động hóa: Tác tử di động giúp tự động hóa các quy trình quản lý mạng, giảm thiểu sự can thiệp của con người.

4. Khả năng mở rộng: Tác tử di động có thể dễ dàng mở rộng để quản lý các hệ thống mạng lớn và phức tạp.

2.14.3 Ứng dụng mô hình tác tử di động CNMMA trong quản lý hệ thống mạng trên Đám mây

1. Giám sát hiệu suất hệ thống

- Thu thập dữ liệu: Tác tử di động di chuyển giữa các máy chủ ảo (VM) hoặc container để thu thập thông tin về hiệu suất như:

- Sử dụng CPU, bộ nhớ, và lưu trữ.

- Băng thông mạng và độ trễ.

- Trạng thái các dịch vụ đang chạy.

- Phân tích dữ liệu: Tác tử di động phân tích dữ liệu tại chỗ hoặc gửi về máy chủ trung tâm để đưa ra cảnh báo hoặc đề xuất tối ưu hóa.

2. Phát hiện và xử lý sự cố

- Phát hiện sự cố: Tác tử di động có thể phát hiện các sự cố như:

- Quá tải tài nguyên (CPU, bộ nhớ, băng thông).

- Lỗi dịch vụ hoặc ứng dụng.

- Sự cố kết nối mạng.

- Tự động xử lý: Tác tử di động có thể tự động thực hiện các hành động như:

+ Khởi động lại dịch vụ.

+ Chuyển hướng lưu lượng.

+ Mở rộng hoặc thu hẹp tài nguyên (scaling).

3. Quản lý tài nguyên động

- Tối ưu hóa tài nguyên: Tác tử di động phân tích việc sử dụng tài nguyên và đề xuất điều chỉnh:

- Tăng hoặc giảm số lượng máy chủ ảo (VM) dựa trên nhu cầu.

- Phân bổ lại băng thông hoặc lưu trữ.

- Tự động scaling: Tác tử di động có thể kích hoạt tự động scaling (horizontal hoặc vertical) để đáp ứng nhu cầu tải.

4. Bảo mật và phát hiện xâm nhập

- Giám sát an ninh: Tác tử di động thu thập dữ liệu liên quan đến an ninh mạng, chẳng hạn:

- Log hệ thống và ứng dụng.

- Lưu lượng mạng đáng ngờ.

- Truy cập trái phép.

- Phát hiện xâm nhập: Tác tử di động phân tích dữ liệu để phát hiện các hoạt động đáng ngờ như:

- Tấn công DDoS.

- Quét cổng (port scanning).

- Phát tán mã độc.

- Phản ứng tự động: Tác tử di động có thể tự động chặn địa chỉ IP đáng ngờ hoặc đóng cổng dịch vụ bị tấn công.

5. Quản lý cấu hình và cập nhật

- Cập nhật cấu hình: Tác tử di động di chuyển giữa các máy chủ ảo để cập nhật cấu hình hệ thống hoặc ứng dụng.

- Triển khai phần mềm: Tác tử di động có thể được sử dụng để triển khai phần mềm hoặc bản cập nhật trên nhiều máy chủ ảo một cách tự động.

6. Thu thập và phân tích dữ liệu lớn (Big Data)

- Thu thập dữ liệu phân tán: Tác tử di động di chuyển giữa các nút trong hệ thống Đám mây để thu thập dữ liệu từ nhiều nguồn khác nhau.

- Phân tích tại chỗ: Tác tử di động có thể phân tích dữ liệu tại chỗ để giảm tải cho mạng và tăng tốc độ xử lý.

7. Hỗ trợ quản lý đa đám mây (Multi-Cloud)

- Di chuyển giữa các Đám mây: Tác tử di động có thể di chuyển giữa các nền tảng Đám mây khác nhau (ví dụ: AWS, Azure, Google Cloud) để quản lý tài nguyên và dịch vụ.

- Đồng bộ hóa dữ liệu: Tác tử di động giúp đồng bộ hóa dữ liệu và cấu hình giữa các hệ thống Đám mây khác nhau.

2.14.4 Lợi ích của việc sử dụng Tác tử di động trong quản lý mạng Đám mây

- Tính linh hoạt: Tác tử di động có thể di chuyển giữa các máy chủ ảo hoặc container để thực hiện nhiệm vụ một cách linh hoạt.

- Giảm tải mạng: Tác tử di động xử lý dữ liệu tại chỗ, giảm lượng dữ liệu cần truyền tải qua mạng.

- Tự động hóa: Tác tử di động giúp tự động hóa các quy trình quản lý mạng, giảm thiểu sự can thiệp của con người.

- Khả năng mở rộng: Tác tử di động có thể dễ dàng mở rộng để quản lý các hệ thống Đám mây lớn và phức tạp.

2.15. Kết luận chương 2

Nội dung chương giới thiệu về công nghệ Tác tử di động, kiến trúc nền tảng tác tử di động và ứng dụng Tác tử di động vào trong thực tế và đề xuất mô hình quản lý mạng CNMMA và quy trình ứng .

Mô hình quản lý mạng điện toán đám mây là Mô hình CNMMA giúp quản lý hiệu quả lưu lượng đám mây với kết quả chính xác hơn và cung cấp bảo mật cho việc quản lý mạng.

Nội dung của chương được tổng hợp và là kết quả của công trình đã được công bố tại [CT1], [CT4] và [CT5]. Các công trình nghiên cứu đem lại hiệu quả và đóng góp khoa học chính như sau:

Thứ nhất, nghiên cứu và đưa ra bộ điều hợp cho phép kết hợp cả hai giao thức SNMP và CMIP cũng như đưa bộ kết nối này vào hai giao thức này vào ứng dụng trong nền tảng tác tử di động.

Thứ hai, xây dựng mô hình quản lý mạng cho mạng cục bộ và đám mây sử dụng nền tảng tác tử di động là Mô hình CNMMA giúp quản lý hiệu quả lưu lượng đám mây với tiết kiệm chi phí, băng thông mạng.

CHƯƠNG 3 NÂNG CAO AN TOÀN BẢO MẬT CHO TÁC TỬ DI ĐỘNG

Trên cơ sở mô hình quản lý mạng CNMMA đã trình bày tại Chương 2, chương này tập trung giải quyết vấn đề bảo mật cho hệ thống mạng cục bộ và đám mây, đặc biệt khi sử dụng nền tảng tác tử di động tuân thủ chuẩn FIPA. Mặc dù tác tử di động mang lại tính linh hoạt và khả năng thực thi phân tán, nhưng chính đặc tính di động và tương tác qua nhiều thực thể trung gian làm gia tăng nguy cơ tấn công. Do đó, việc cải tiến các cơ chế bảo mật cho nền tảng này là yêu cầu cấp thiết để có thể ứng dụng vào thực tiễn.

Để hiện thực hóa các yêu cầu bảo mật đã được xác định trong mô hình CNMMA (bí mật, toàn vẹn, xác thực, kiểm soát truy cập), hai giải pháp chính được đề xuất như sau:

Giải pháp thứ nhất, cải tiến kỹ thuật trên nền tảng tác tử di động như cải thiện cơ chế bảo mật nền tảng mã hóa trạng thái tác tử khi di chuyển, tối ưu hóa quy trình thực thi tác tử trong môi trường ứng dụng, giới hạn vòng đời tác tử, để giảm thiểu tác động khi tác tử bị xâm nhập…Đồng thời, các cải tiến này được thiết kế dưới dạng module mở rộng, không phá vỡ chuẩn FIPA.

Giải pháp thứ hai, xây dựng mô hình phát hiện xâm nhập (IDS - Intrustion Detection System) tích hợp trên đám mây, nhằm bảo vệ toàn bộ hệ thống mạng trước các mối đe dọa từ bên ngoài cũng như từ chính nền tảng và tác tử di động.

Hai nhóm giải pháp này tích hợp chặt chẽ với nhau: các cải tiến trên nền tảng tác tử tạo thành lớp bảo vệ từ bên trong (mã hóa, xác thực, kiểm soát truy cập), trong khi mô hình IDS trên đám mây tạo thành lớp bảo vệ từ bên ngoài (giám sát, phát hiện, phản ứng). Nhờ đó, hệ thống vừa đảm bảo an toàn cho tác tử di động, vừa bảo vệ được hạ tầng mạng cục bộ và đám mây, đồng thời vẫn giữ được tính linh hoạt, mềm dẻo và khả năng mở rộng theo đúng tinh thần của mô hình CNMMA đã đề xuất tại Chương 2.

3.1 Nguy cơ bảo mật trên công nghệ tác tử di động

Không thể phủ nhận lợi ích và tính thích nghi, tính di động, hoạt động mềm dẻo của công nghệ tác tử di động. Tuy nhiên, sự năng động của tác tử di động dẫn tới sự phức tạp trong thiết kế và gây ra một số nguy cơ về bảo mật. Mối đe doạ về an toàn bảo mật có thê được phân ra thành bốn hạng mục cơ bản sau [24]:

1. Tác tử tới nền tảng (**Agent to Platform)**: Hạng mục này liên quan đến mối đe doạ của tác tử tới một nền tảng cụ thể.
2. Nền tảng tới tác tử (**Platform to Agent):** Hạng mục này liên quan đến mối đe doạ nền tảng tới tác tử
3. Tác tử tới tác tử (**Agent to Agent):** Hạng mục này liên quan đến mối đe doạ giữa các tác tử với nhau
4. Nền tảng tới nền tảng (**Platform to Platform):** Hạng mục này liên quan đến mối de doạ giữa các nền tảng chứa các tác tử

3.2 Hình thức tấn công tác tử di động

Các hình thức tấn công này xếp vào các hạng mục theo bảng 3.1 như sau:

Bảng 3.1: Các hình thức tấn công theo mối đe doạ [24]

	Hạng mục mối đe doạ			
Hình thức tấn công	***1***	***2***	***3***	***4***
Giả mạo (Masquerading)	Có	Có	Có	Có
Truy cập trái phép (Unauthorized Access)	Có	Có	Có	Không
Từ chối dịch vụ (Denial of Service)	Có	Có	Không	Không
Chối bỏ (Repudiation)	Có	Có	Không	Không
Sự thay thế (Alternation)	Không	Có	Có	Không
Nghe lén (Eavesdropping)	Không	Có	Không	Không

Sự giả mạo (Masquerading)

Tấn công giả mạo xảy ra khi một người dùng bất hợp pháp giả dạng người dùng hợp pháp với mục đích lấy cắp quyền của người dùng hợp pháp để thực hiện các hoạt động không được phép. Kiểu tấn công này xuất hiện cả 4 hạng mục mối đe doạ [26].

Trong hạng mục từ nền tảng với nền tảng và nền tảng tới tác tử, các tác tử không được xác thực có thể cướp nhận dạng của một tác tử khác và do đó, lấy được quyền truy cập vào dịch vụ và tài nguyên mà nó không được cấp. Cũng tương tự, một nền tảng tác tử có thể giả mạo một nền tảng khác và đón lấy các tác tử di động thay vì một đích tới hợp lệ. Điều này cho phép một nền tảng giả có thể trích xuất thông tin nhạy cảm từ các tử khi vào trong nền tảng [24].

Các tác tử có thể kết nối với nhau và do đó các tác tử độc hại có thể lừa dối các tác tử khác và trích xuất thông tin nhạy cảm mà nó lấy được. Với cách tương tự với một nền tảng, một nền tảng giả mạo cũng có thể lừa dối một tác nền tảng khác và yêu cầu trao đổi thông tin giữa các tác tử và yêu cầu các tác tử của nền tảng chuyển sang nền tảng giả mạo [24].

Truy cập trái phép (Unauthorized Access)

Kiểu tấn công này thường được áp dụng cho hạng mục tác tử tới tác tử và tác tử tới nền tảng, và nền tảng tới tác tử. Bằng việc tấn công tác tử di động để lấy quyền truy cập tới một nền tảng hợp lệ và gây hậu quả tới các tác tử hợp lệ khác. Bên cạnh đó, một tác tử di động có thể gây gián đoạn trực tiếp tới tác tử khác bằng việc triệu gọi các Phương thức công khai. Thậm chí, nó có thể truy cập, thay đổi dữ liệu hay cả mã nguồn của tác tử. Sự thay đổi đó gây ra hậu quả là làm thay đổi các hành vi của tác tử hợp lệ (biến một tác tử tin cậy thành một tác tử độc hại) [24].

Tấn công từ chối dịch vụ (Denial of Service)

Thông thường, khái niệm tấn công từ chối dịch vụ được sử dụng cho các tấn công làm cạn kiệt nguồn tài nguyên gây ra hậu quả là các thực thể khác không thể thực hiện được yêu cầu. Khái niệm này cũng vẫn hiện hữu trên nền tảng tác tử. Tuy nhiên, trong hệ thống nền tảng tác tử, định nghĩa này được mở rộng để ngăn chặn một tác tử tiếp tuc di chuyển đến một nền tảng khác hoặc thậm chí xoá bỏ một tác tử.

Kiểu tấn công này được áp dụng cho nền tảng tới tác tử, tác tử tới tác tử, và tác tử tới nền tảng[24].

Chống chối bỏ (Repudiation)

Kiểu tấn công chối bỏ ám chỉ sự từ chối trách nhiệm trong việc tham gia vào các kết nối và giao dịch. Sự chối bỏ có thể dẫn đến tranh chấp nghiêm trọng và không dễ giải quyết nếu các biện pháp phòng chống không được áp dụng [24].

Sự thay thế (Alternation)

Hạng mục mối đe doạ từ nền tảng tới tác tử là mục tiêu của kiểu tấn công này. Kiểu tấn công này ám chỉ sự thay thế của mã độc vào trong mã nguồn hoặc dữ liệu của các tác tử mà không bị phát hiện.

Sự phát hiện thay thế của mã độc không đơn giản và hiện nay cũng chưa có giải pháp chung cho vấn đề này bởi mỗi nền tảng cũng cần phải có quyền truy cập tới một phần mã nguồn, dữ liệu của tác tử và hậu quả là có thể thay đổi chúng [24].

Nghe lén (Eavesdropping)

Đây là một hình thức tấn công bị động cho phép chặn và theo dõi các kênh truyền thông bí mật giữa các tác tử. Đối với các tử di động, nghe lén càng dễ dàng hơn bởi nền tảng có thể theo dõi mọi chỉ thị lệnh mà tác tử thực thi. Hình thức này cần phân tích rất mạnh về hành vi của tác tử [24]

3.2.1 Một số mô hình mối đe doạ (threat model) khi ứng dụng nền tảng tác tử di động

- Giả mạo tác tử (Agent Spoofing)

+ Mô tả: Kẻ tấn công tạo một tác tử độc hại giả mạo danh tính tác tử hợp pháp để gửi lên nền tảng hoặc giao tiếp với các tác tử khác.

+ Nguy cơ: Truy cập trái phép dữ liệu nội bộ; Kích hoạt hành động thay mặt tác tử thật.

+ Điều kiện khai thác: Thiếu cơ chế xác thực mạnh giữa tác tử và nền tảng; Khóa bí mật (private key) của tác tử bị lộ hoặc yếu.

- Sửa mã tác tử (Code Tampering)

+ Mô tả: Thay đổi mã nguồn hoặc bytecode của tác tử trước hoặc trong khi tác tử được triển khai.

+ Nguy cơ: Chèn logic độc hại (đánh cắp dữ liệu, phá hoại);Vô hiệu hóa cơ chế an toàn của tác tử.

+ Điều kiện khai thác: Không có chữ ký số trên mã tác tử; Nền tảng không kiểm tra toàn vẹn mã khi tải hoặc thực thi.

- Tấn công phát lại (Replay)

+ Mô tả: Ghi lại một giao tin (message) hợp lệ giữa tác tử và nền tảng, sau đó phát lại để đánh lừa.

+ Nguy cơ: Thực hiện lại hành động cũ (ví dụ: chuyển tiền, thay đổi trạng thái).

+ Điều kiện khai thác: Thiếu nonce, timestamp hoặc sequence number; Không kiểm tra tính tươi (freshness) của giao tin.

- Tấn công xen giữa MITM (Man-in-the-Middle)

+ Mô tả: Kẻ tấn công chặn, sửa hoặc chèn giao tin giữa tác tử và nền tảng.

+ Nguy cơ: Đánh cắp thông tin xác thực; Thay đổi lệnh hoặc dữ liệu đang truyền.

+ Điều kiện khai thác: Kênh truyền không được bảo vệ (không TLS hoặc TLS tự ký không kiểm tra); Thiếu chứng thực lẫn nhau (mutual authentication).

3.2.2 Phương thức bảo mật cơ bản cho nền tảng tác tử

Phương thức bảo mật cơ bản cho hệ thống tác tử di động cần yêu cầu các tính chất sau: tính xác thực, tính bảo mật, tính sẵn sàng, trách nhiệm và tính chống chối bỏ.

Xác thực và phân quyền truy cập

Xác thực là tiến trình kiểm tra về định danh của một thực thể. Trong hệ thống tác tử di động, tiến trình xác thực cần đòi hỏi cả tác tử và nền tảng cần phải được xác thực bởi môi trường thực thi có hiểu biết về tác tử. Phân quyền truy cập là tiến trình quyết định gán quyền hoặc không cho một thực thể đã được xác thực. Để đạt được tính chất về bảo mật, chữ kí số hoặc bảo vệ bằng mật khẩu có thể được sử dụng kết hợp [26].

Tính bảo mật, Tính riêng tư và tính ẩn danh

Tính bảo mật ám chỉ tới trạng thái ẩn các thông tin nhạy cảm đối với các đối tượng không được xác thực. Sự tiết lộ các thông tin này có thể làm giảm mức độ riêng từ bởi dữ liệu này có thể chứa các thông tin riêng tư liên quan tới tác tử. Bộc lộ các hành vi của tác tử di động có thể giảm mức độ riêng tư đối với một số mở rộng. Sự quan tâm tới tính riêng tư có thể được đáp ứng qua kĩ thuật đảm bảo tính riêng như thiết lập cấp độ ẩn danh đối với các đối tượng. Mã hoá dữ liệu cũng là là một trong những Phương thức tốt trong việc ẩn các dữ liệu nhạy cảm đối với các đối tượng không được xác thực tuy nhiên về mặt hiệu của hệ thống có thể sẽ bị suy giảm [24].

Trách nhiệm và chống chối bỏ

Những vấn đề liên quan tới sự chối bỏ sẽ tăng lên khi một đối tượng phủ định rằng có tham gia trong một hoạt động hoặc kết nối nào đó trong khi đó thực sự đã tham gia. Để giải quyết vấn đề trên, sự quan trọng của truyền thông và các hoạt động liên quan đến bảo mật cần phải được ghi lại để kiểm tra và lần vết nhằm mục đích chống chối bỏ trách nhiệm. Các dữ liệu nhật kí này cần phải được bảo vệ đối với những đối tượng không được phép để duy trì tính riêng tư và cấp độ bảo mật cho hệ thống

Tính sẵn sàng

Một nền tảng tác tử di động cần đảm bảo chắc chắn sự sẵn sàng của dữ liệu và dịch vụ được cung cấp bởi các tác tử di động. Điều này ám chỉ nền tảng cần phải cung cấp khả năng điều khiển đồng thời, truy cập đồng thời, quản lý deadlock và phân quyền truy cập theo yêu cầu. Các nền tảng cần phải có thể khả năng phát hiện và phục hồi các phần mềm khi bị treo như phần cứng bị lỗi. Đồng thời, nó cũng cần phải xử lý tốt đối với các tình huống tấn công từ chối dịch vụ DoS [24].

3.3 Cách phòng chống nguy cơ đe doạ bảo mật

Một số cơ chế và kỹ thuật đưa ra để phát hiện và ngăn chặn tấn công trên hệ thống tác tử di động. Những cơ chế này được sử dụng để đảm bảo các nền tảng này sẽ tuân theo các chính sách để phục vụ các tác tử di động.

Kỹ thuật phát hiện xâm nhập

Kỹ thuật phát hiện xâm nhập được sử dụng để phát hiện ra khi một tác tử có sự thay đổi hoặc không. Nó bao gồm giả mạo mã nguồn, trạng thái, luồng thực thi. Có

nhiều kỹ thuật khác nhau liên hệ tới việc chúng có thể tự động hoặc không tự động, thực thi công việc trong quá trình làm việc hoặc huỷ, phát hiện một phần hoặc toàn bộ sự thay đổi.

Cơ chế phát hiện xâm nhập cũng rất phong phú dựa trên phạm vi việc phát hiện, một số kỹ thuật sử dụng khoảng kiểm tra để phát hiện thao tác thay đổi mã nguồn bất hợp pháp theo như các giá trị của biến hoặc ràng buộc về mặt thời gian. Sự lần vết các thực thi và mã hoá cho phép chúng có thể phát hiện tấn công vào mã nguồn, trạng thái, luồng thực thi trên các thành phần của tác tử. Hàm băm (hash) cũng được đề cử như là một giải pháp để bảo vệ tính toàn vẹn về thông tin của các phần mềm tác tử[26].

Kỹ thuật ngăn chặn

Những cơ chế ngăn chặn thường được sử dụng để làm tăng lên các cấp độ bảo mật của tác tử di động chống lại các cuộc tấn công giả mạo. Các kỹ thuật này được sử dụng để đảm bảo mức độ khó trong việc truy trái phép hoặc thay đổi mã nguồn. Kỹ thuật ngăn chặn cũng rất phong phú tuỳ thuộc vào mục đích ngăn chặn: Ngăn chặn toàn bộ tác tử hoặc thành phần của nó, ngăn chặn tấn công vĩnh viễn hoặc tạm thời, sử dụng chức năng tin cậy hoặc giả định sự không tin cậy [24].

Một số kỹ thuật khác phụ thuộc vào môi trường tin cậy được trang bị phần cứng chống giảm mạo đối với mỗi nền tảng. Tuy nhiên, sự mở rộng phần cứng chống giả mạo đối với mỗi nền tảng làm giới hạn khả năng của tác tử di động đối với các tính năng của tác tử. Sử dụng các chức năng mã hoá để ngăn chặn giả mạo. Xu hướng mã hoá đối với cả mã nguồn và dữ liệu thông tin trạng thái của tác tử bằng cách sử dụng điện toán trực tiếp mã hoá trên dữ liệu mà không cần giải mã [24].

Việc sử dụng mã hoá đồng nhất là một của dữ liệu sẽ được thực thi trên máy chứa tác tử mà không cần giải mã. Ý tưởng chính của giải pháp này là sử dụng máy chủ trung tâm chịu trách nhiệm mã hoá và giải mã. Hay thiết kế bộ giao thức bảo mật cho nền tảng tác tử để truyền thông tin và dữ liệu, thực hiện chức năng một cách an toàn và bảo mật trong môi trường không an toàn và tin cậy.

3.4 Bảo vệ tính toàn vẹn trong công nghệ tác tử di động

Hàm mã hoá

Hàm mã hoá EF (Encrypted Function) là một bước chuyển tiếp để cài đặt bảo mật cho hộp đen hoàn hảo. Mục đích của các hàm mã hoá là để nhận biết được phương thức nào sẽ được cho phép mã nguồn di động được mã hoá qua thuật toán điện toán như chữ kí số, hoặc thậm chí mã nguồn có thể thực thi trong môi trường không tin cậy và hoạt động các thao tác tự động mà không cần sự tương tác của môi trường nền tảng gốc. Xu hướng này cho phép nền tảng tác tử có thể thực thi các chương trình mô phỏng các hàm mã hoá mà không cần phải trích xuất về dạng gốc. Xu hướng này cũng hoàn toàn khác biệt so với các hàm và chương trình cài đặt các hàm [24].

Một hệ thống EF được mô tả bao gồm:

A là một thuật toán để tính tính chức năng f. B là đầu vào các giá trị x và sẽ được tính toán qua hàm $f(x)$ cho thuật toán A, nhưng A muốn B không cần quan tâm tới thành phần bên trong f. Hơn nữa, B không muốn tương tác với A trung suốt quá trình thực hiện hàm $f(x)$.

Để cài đặt hệ thống thực hiện việc trên, chúng ta cần phải giả sử rằng hàm f có thể mã hoá một hàm khác $E(f)$. Tiếp theo, lược đồ có thể được xây dựng như sau:

- A mã hoá f và lấy hàm $E(f)$,
- A tạo ra chương trình $P(E(f))$ mà cài đặt $E(f)$
- A gửi $P(E(f))$ tới B,
- B thực thi $P(E(f))$ trên x,
- B gửi kết quả chương trình $(P(E(f))(x))$ đến A,
- A giải mã và nhận kết quả qua hàm $f(x)$

Hàm f có thể là các thuật toán như chữ kí số với việc nhúng khoá hoặc các thuật toán mã hoá chứa một trong số chúng. Điều này có thể cho phép các tác tử được kí và mã hoá dữ liệu tại ngay máy chủ host mà không cần phải lấy khoá bí mật.

Mặc dù ý tưởng này là rất dễ cài đặt nhưng chúng lại khó để tìm ra mô hình mã hoá có thể thực thi các hàm như vậy. Công nghệ cho phép mã hoá các hàm và đa thức đã được đề xuất, và các giải pháp mã hoá này có thể sử dụng hệ thống mã hoá RSA

Bảo mật hộp đen giới hạn thời gian và làm xáo trộn mã nguồn

Thực chất hộp đen bảo mật là khó cài đặt, bởi hiện tại chưa thể thực hiện được, như giả sử hộp đen không thể đảm bảo được mãi mãi, nhưng có thể đảm bảo trong khoảng thời gian. Theo như định nghĩa, một tác tử có thuộc tính hộp đen giới hạn thời gian cho phép trong một khoảng thời gian nào đó hộp đen là bất khả xâm phạm [26].

Ý tưởng chính của hướng này là sinh ra các tác tử có thể thực thi từ các đặc tả tác tử được đưa ra, và không thể bị tấn công bằng cách chỉ đọc hoặc thao tác. Hộp đen giới hạn thời gian đáp ứng thuộc tính hộp đen trong khoảng thời gian giới hạn bằng cách:

- Mã nguồn và dữ liệu đặc tả của tác tử không thể đọc
- Mã nguồn và dữ liệu đặc tả của tác tử không thể bị thay đổi

Điều này đồng nghĩa với việc dữ liệu không thể thêm vào, mặc dù các biến là có tồn tại và có thể thay đổi.

Để đạt được thuộc tính hộp đen, các thuật toán chuyển đổi được đề xuất sử dụng là các thuật toán làm xáo trộn hoặc làm trở thành rắc rối và phức tạp để có thể đọc được dữ liệu và mã nguồn. Các thuật toán này sinh ra các tác tử mới từ các tác tử gốc mặc dù có mã nguồn khác nhưng cùng đưa ra một kết quả giống nhau.

Phương thức làm xáo trộn mã nguồn khá phức tạp để đọc hiểu được mã nguồn. Để thay đổi mã nguồn của chương trình cần phải sử dụng tham số và tự động hoá để biến các mã nguồn có thể đọc được thành không thể [26].

Các tham số bổ sung có thể sử dụng để làm cho khả thi các chương trình được làm xáo trộn giống với các chương trình nguyên gốc. Sự phức tạp trong việc chuyển đổi các chương trình là một cách để các chương trình không dễ dàng dịch ngược trở lại. Một vấn đề khác nảy sinh là rất khó đo lường được chất lượng của sự xáo trọng, không chỉ phụ thuộc vào thuật toán được sử dụng mà còn phụ thuộc vào khả năng của kĩ thuật dịch ngược.

Bởi một tác tử có thể trở thành không hợp lệ trước khi chuyển đổi hoàn toàn và xáo trộn mã nguồn chỉ phù hợp với các ứng dụng không kèm theo dữ liệu. Biện pháp này có thể xáo trộn mã nguồn nhưng dữ liệu thì không bị thay đổi nên kẻ tấn công

vẫn có thể đọc và thao tác với dữ liệu, tuy nhiên kết quả này cũng ngẫu nhiên và có thể không có giá trị đối với kẻ tấn công.

Dấu vết mật mã (Cryptographic Trace)

Giovanni Gigna giới thiệu dấu vết mật mã (cũng được gọi là dấu vết thực thi) là cách cung cấp việc kiểm tra sự đúng đắn trong thực thi tác tử. Phương thức này dựa trên dấu vết của thực thi tác tử, cung cấp các yêu cầu từ bộ tác tử gốc sau khi tác tử được dừng và sử dụng làm cơ sở cho việc kiểm tra thực thi. Kĩ thuật này yêu cầu mỗi nền tảng bao gồm việc tạo và duy trì sự chống chối bỏ trong các nhật kí hoặc dấu vết các hoạt động được thực thi bởi mỗi tác tử mà nó lưu trú, và thực hiện mã hoá bảng băm theo dấu vết dựa trên các kết luận giống như tổng hợp dấu vết và dấu vân tay. Dấu vết được cấu thành từ chuỗi các định danh chỉ thị lệnh và thông tin chữ kí nền tảng. Chữ kí nền tảng cần thiết cho các chỉ thỉ lệnh phụ thuộc vào sự tương tác giữa các môi trường điện toán được duy trì trên nền tảng. Với các chỉ thị lệnh chỉ phụ thuộc vào giá trị thì các biến nội tại, không cần một chữ kí nào do đó nó bị bỏ qua.

Cơ chế này cho phép phát hiện tấn công vào mã nguồn, trạng thái và luồng điều khiển tác tử di động. Bằng cách này, trong trường hợp giả mạo, chủ sở hữu tác tử có thể chứng minh các hoạt động mà không bao giờ được thực hiện bởi tác tử.

Kĩ thuật này cung cấp định nghĩa các giao thức bảo mật để chuyên trở các các tác tử và đi kèm các thông tin bảo mật đi kèm trong vài nhà cung cấp, có thể bao gồm các nhà cung cấp đáng tin cậy để duy trì chuỗi tổng hợp các dấu vết cho toàn bộ các tác tử ban đầu. Nếu có những kết quả đáng ngờ, dấu vết phù hợp và các tổng hợp dấu vết có thể lấy được và kiểm tra, và xác định được nền tảng mã độc [24].

Xu hướng này có một số nhược điểm, trong đó nhược điểm chính là kích thước và số lượng nhật kí để duy trì và thực tế là các tiến trình phát hiện được kích hoạt thì số lượng lớn kết quả cần phải giám sát và so sánh. Một nhược điểm khác là việc xác định định danh bao hàm sự thiếu các tác tử đa tiến trình phù hợp và các kỹ thuật tối ưu động. Trong khi mục tiêu của kỹ thuật này là bảo vệ tác tử, kỹ thuật cũng đưa ra một số sự bảo vệ nền tảng tác tử, cung cấp nền tảng có lấy được tổng hợp về dấu vết và dấu vết từ các tổ chức khác.

Giao thức mắt xích MAC (chained MAC)

Giao thức mắt xích MAC có nhiều phiên bản tồn tại, một chúng tồn tại dựa trên nền tảng mã hoá công khai, một số khác dựa trên khoá đơn. Giao thức này cho phép một tác tử có thể đạt được tính toàn vẹn đầy đủ nhất. Để tối ưu cho giao thức này, chỉ có khoá công khai của nguyên gốc được các nền tảng nhận biết được [24].

Giả sử, r_n là một số ngẫu nhiên được sinh ra bởi mỗi máy chủ host. Các giá trị được sử dụng giống như mã bí mật trong MAC (Message Authentication Code). Thành phần kết quả m_n, mẫu ngẫu nhiên r_n và xác định thành phần máy chủ host tiếp theo được mã hoá với mã khoá công khai của nguyên gốc $K(i_0)$, được đóng gói thành thông điệp M:

$$M\,n \;=\; \{rn\,, mnid\;(i+l\,)\;Ki0$$

Chuỗi quan hệ được định nghĩa theo (Kí hiệu H là hàm băm):

$$hO \;=\; \{rn\,, mOid\;(iO\;)\;\}\;Ki0$$

$$hn \;=\; H\;(\;hn-l\,, rn\,, on\,, id\;(in+1\;))$$

Khi tác tử di trú từ máy chủ in tới i_{n+1}:

$$in \;\rightarrow\; in+l\;:\;\{M\;k\;|O\;\leq\;k\;\leq\;n\}, hn$$

Dấu mờ (Watermark)

Dấu mờ thường được sử dụng để bảo vệ bản quyền nội dung số. Một nhà phân phối hoặc chủ sở hữu nội dung sẽ nhúng dấu vào nội dung đó, vì vậy quyền sở hữu có thể được chứng minh. Thông thường chúng có thể giấu kín. Các phương thức khai thác thông tin có thể dư thừa, nhưng sử dụng giải pháp này có thể bảo vệ dữ liệu và mã nguồn tác tử di động.

Một trong những Phương thức của dấu mờ là một dấu tích được nhung vào trong tác tử di động sử dụng kỹ thuật dấu mờ phần mềm. Dấu tích này được truyền tới kết quả của tác tử trong suốt thời gian thực thi. Đối với máy chủ chứa tác tử thì dấu tích này được coi là một phần của kết quả và cũng dường như "vô hình". Nếu chủ sở hữu của tác tử phát hiện ra dấu tích đã bị thay đổi (khác với kết quả mong đợi) thì họ có thể chứng minh được máy chủ chứa tác tử đã thay đổi dữ liệu hoặc mã nguồn của tác tử [26].

Để tạo một dấu tích cho tác tử, có thể sử dụng các cách thức sau:

- Đánh dấu tích vào trong mã nguồn

- Đánh dấu tích vào trong dữ liệu đầu vào

- Đánh dấu tích vào trong mã nguồn đã làm xáo trộn

- Dấu tích hoặc nhiều dấu tích cần được kiểm tra sau khi tác tử đã phục hồi lại trạng thái (sau khi di trú).

- Một số hình thức tấn công vào cách phòng chống này:

- Nghe lén (eavesdropping): Nếu dữ liệu không được bảo vệ bằng các Phương thức khác nhau (ví dụ: không mã hoá) có thị bị đọc bởi bất cứ máy chủ host nào.

- Tác động (manipulation): máy chủ nhiễm độc có thể cố gắng tác động và mã nguồn hoặc dữ liệu của tác tử để thay đổi kết quả mà vẫn giữ nguyên các dấu tích.

- Sự cấu kết: một nhóm các máy chủ host bị nhiễm có thể cấu kết với nhau để khám phá dấu tích bằng việc so sánh kết quả lấy được.

Dấu vân tay

Dấu vân tay bằng phần mềm sử dụng một phần mềm sử dụng kĩ thuật dấu mờ để nhúng dấu tích của mỗi người dùng. Dấu vân tay phần mềm chia sẻ tính chất với các phần mềm nhúng dấu mờ khác: dấu tích cần phải được thao tác thầm lặng và "vô hình" đối với các hệ thống theo dõi [24].

Trong phương thức sử dụng dấu vân tay, đối lập với phương thức dấu mờ, dấu tích của mỗi máy chủ host đều được in lại. Khi tác tử trở lại chủ sở hữu, mọi kết quả đều được kiểm tra lại. Do đó, các máy chủ bị nhiễm mã độc sẽ có vết trực tiếp.

Trong Phương pháp sử dụng vân tay cho tác tử di động, dấu tích nhúng sẽ khác với mỗi máy chủ host. Cách thực hiện nhúng bằng dấu mờ cũng giống với Phương thức vân tay.

Sự khác biệt giữa hai phương thức này có khả năng phát hiện sự cấu kết thực hiện tấn công bởi nhóm các máy chủ không tin cậy.

Ba cách để nhúng vân tay vào trong tác tử:

- Dấu trong mã nguồn: trong trường hợp này, máy chủ nhiễm độc có khả năng so sánh giữa các đoạn mã để tìm ra dấu tích

- Dấu trong dữ liệu đầu vào: dữ liệu thường được sử dụng khác với mỗi máy chủ host nên rất khó để phát hiên ra dấu tích này.

- Đánh dấu tích vào trong mã nguồn đã làm xáo trộn

Các cách thức thực hiện cũng đều giống với dấu mờ. Tuy nhiên, các máy chủ host nguyên thuỷ cần phải biết được dấu tích của mỗi máy chủ host và vị trí của chúng. Một trong những khả năng đó là tái tạo lại dấu tích bằng việc lấy thông tin từ các nơi trước đó trong kết quả.

Một số hình thức tấn công vào cách phòng chống này:

- Nghe lén (eavesdropping): Nếu dữ liệu không được bảo vệ bằng các Phương thức khác nhau (ví dụ: không mã hoá) có thị bị đọc bởi bất cứ máy chủ host nào.

- Tác động (manipulation): máy chủ nhiễm độc có thể cố gắng tác động và mã nguồn hoặc dữ liệu của tác tử để thay đổi kết quả mà vẫn giữ nguyên các dấu tích.

- Sự cấu kết: một nhóm các máy chủ host bị nhiễm có thể cấu kết với nhau để khám phá dấu tích bằng việc so sánh kết quả lấy được.

Một số giao thức khác

Chuỗi chữ kí số có thể kiểm tra công khai (Publicly Verifiable Chained Digital Signatures)

Giao thức này cho phép kiểm tra các chuỗi kết quả của tác tử khong chỉ ở tại host nguyên thuỷ mà còn ở bất cứ nơi nào tác tử hoạt động. Tuy nhiên, nó vẫn tồn tại lỗ hổng bị tấn công. Giao thức này giúp cho các tác tử đang cư trú có thể nhận được sự kiểm tra để đánh dấu không bị giả mạo. Điều này giúp cho tiết kiệm năng lực tính toán bởi trong trường hợp tác tử bị giả mạo thì máy chủ chứa tác tử sẽ từ chối thực hiện các tác vụ của tác tử giả mạo [24].

Sinh khoá cho môi trường (Environmental Key Generation)

Lược đồ này cho phép các tác tử có thể định nghĩa trước các hoạt động với một số điều kiện môi trường đúng. Hướng hoạt động của giải pháp này là xây dựng tác tử theo cách ứng xử với các điều kiện của môi trường (ví dụ: so khớp với chuỗi tìm kiếm), sinh ra một khoá được sử dụng để mở khoá cho thực thi đoạn mã hoá. Điều kiện môi trường bị ẩn theo một chiều băm hoặc mã hoá công khai trong môi trường có khởi tạo. Kĩ thuật này đảm bảo nền tảng hoặc bộ giám sát các tác tử không thể

kiểm soát các thông điệp khởi tạo hoặc các hành động đáp ứng bằng cách trực tiếp đọc mã nguồn của tác tử [24].

Ghi lại các khởi điểm bằng nhân bản và đánh giá (Itinerary Recording with Replication and Voting)

Một tác tử bị lỗi có cách ứng xử tương tự các tác tử bị nhiễm mã độc. Do đó, bằng cơ chế dung lỗi trong môi trường hoạt động có thể nhận biết được sự ảnh hưởng của tác tử đối với nền tảng bị nhiễm độc. Một kĩ thuật cho những Phương thức trên nhằm đảm bảo một tác tử khi tới một nền tảng phải an toàn là sử dụng nhân bản và đánh giá.

Ý tưởng này ít được sử dụng hơn là chỉ dùng một phiên bản của tác tử để thực hiện tính toán. Bởi một nền tảng bị nhiễm độc có làm ảnh hưởng tới nhiều bản sao của tác tử, nhưng việc sử dụng các bản sao khác nhau cũng giúp cho việc thực hiện tác vụ được "xuôi chèo mát mái" hơn nhờ có các bản dự phòng [24].

3.5 Bảo mật kênh truyền dẫn trong công nghệ Tác tử di động

3.5.1 Giới thiệu chung

Tác tử di động là phần mềm có khả năng dịch chuyển giữa các máy tính trong hệ thống mạng. Nó có thể thực thi các đoạn mã thực thi trên các máy từ xa và thực hiện giống như đang thực hiện giống như phần mềm được cài đặt trên máy đó. Thuộc tính di động cho phép các tác tử có thể di chuyển và mô hình tác tử di động tạo ra một phương thức hoạt động hoàn toàn mới với các kỹ thuật, sử dụng tài nguyên và ứng dụng mới.

Để các tác tử và nền tảng cho tác tử cho thể truyền dẫn các thông điệp cho nhau thì giao thức truyền vận tác tử MTP (Message Tranfer Protocol) là rất cần thiết. Đồng thời, yếu tố bảo mật đường truyền, đặc tả cho kênh truyền dẫn tác tử bảo mật đặc biệt quan trọng giúp cho hệ thống quản lý dựa trên tác tử di động an toàn và bảo mật hơn [38].

3.5.2 Giao thức truyền vận thông điệp cho IIOP

Giao thức truyền vận thông điệp cho IIOP dựa trên cấu trúc OMG IDL chứa bao bì của thông điệp và chuỗi tuần tự octet để biểu diễn nội dung của thông điệp theo

chuẩn ACL. Bao bì và nội dung của thông điệp được truyền đi qua việc triệu gọi IIOP một chiều [37].

Khi một yêu cầu nhận được, thông điệp được đóng gói và được sử dụng bởi ACC để chỉ thị lệnh và các thông tin cần thiết tương ứng để xử lý nội dung thông điệp.

Định nghĩa giao tiếp

IDL sau chỉ ra giao tiếp truyền vận thông điệp. Giao tiếp này chứa hoạt động của thông điệp yêu cầu một đối số. Một đối số có 2 thuộc tính, chuỗi các cấu trúc bao bì chứa bao bì và nội dung thông điệp [38]:

Bảng 3.2: Mã giả Psedocode về định nghĩa giao tiếp IDL

```
module FIPA {
 typedef sequence<Envelope> Envelopes;
 typedef sequence<octet> Payload;
 struct FipaMessage {
   Envelopes messageEnvelopes;
   Payload   messageBody;
 };
 interface MTS {
   oneway void message(in FipaMessage aFipaMessage);
 };
};
```

ACC xử lý cho bao bì IDL

Theo đặc tả FIPA, ACC theo chuẩn FIPA không được phép thay đổi các thành phần đóng gói trong bao bì thông điệp mà nó nhận được. Tuy nhiên, nó cho phép cập nhật các giá trị của một hay nhiều các đối số của bao bì bằng cách thêm các thành phần vào bao bì trong cuối của chuỗi messageEnvelope. Thành phần này là cần thiết để cho phép các gí trị của tham số mà ACC mong muốn được thêm và cập nhập giá trị vào thành phần ReceivedObject [37].

EnvelopeWithAllFields chứa các giá trị mới nhất của các trường dữ liệu.

Bảng 3.3: Mã giả Psedocode về đóng gói bao bì cho ACL [37].

```
EnvelopeWithAllFields := new empty Envelope;
while ((EnvelopeWithAllFields does not contain values for all its fields)
      OR (all Envelopes in the sequence have been processed)) {
 // the ACC gets the next envelope in the sequence starting from the end
 tempEnvelope = getNextEnvelope;
foreach field in an envelope {
   if ((this field has no value in envelopeWithAllFields)
      AND (this field has a value in tempEnvelope))
  then copy the value of this field from tempEnvelope to envelopeWithAllFields
 }
}
```

Bảng 3.4: Mã giả Psedocode về đóng gói bao bì cho ACL chứa trường dữ liệu [37].

```
Envelope(0):
 to = tizio
 from = caio
 aclRepresentation = XML
 received = ...
Envelope (1):
 from = caio@molfetta.it
 received = ...
Envelope (2):
 intended-receiver = tizio@villardora.it
 received = ...
EnvelopeWithAllFields:
to = tizio                          (from envelope 0)
from = caio@molfetta.it             (from envelope 1)
intended-receiver = tizio@villardora.it   (from envelope 2)
date = 25 May 2000                  (from envelope 0)
```

Hệ quả là ACC nhận được thông điệp cần phải được cài đặt các thủ tục mô tả bằng đoạn pseudo-code sau. Thủ tục này sẽ soạn lại cấu trúc bao bì đầy đủ với các giá trị cập nhật mới nhất cho các tham số. Thủ tục cũng đơn gian chỉ ACC bắt đầu từ cuối các chuỗi và tiếp cho tới khi đến giá trị cần thiết đối với mỗi tham số trong bao bì [37].

Cú pháp bao bì thông điệp rời rạc

Cú pháp bao bì trừu tượng theo chuẩn FIPA được ánh xạ từ tập các cấu trúc OMG IDL, được sử dụng trong các module theo chuẩn FIPA.

Đoạn mã sau chỉ ra chuẩn cho việc định danh các tham số tuỳ chọn: Một chuỗi rỗng hoặc định danh của chuỗi tuần tự các chuỗi rỗng tới các tham số. Trong trường hợp các tham số độ dài tải thường là các giá trị âm và có thể sử dụng để xác định với các giá trị không hiện hữu (non-presence) B.

3.5.3 Giao thức truyền vận thông điệp cho HTTP

Giao thức truyền vận thông điệp dựa trên HTTP cho phép biểu diễn dữ liệu của toàn bộ thông điệp tác tử bao gồm cả bao bì của một yêu cầu HTTP. Truyền vận giao thức thông điệp qua HTTP là tiến trình gồm 2 bước: người gửi tạo ra các yêu cầu HTTP và nhận dữ liệu đóng gói và thông điệp theo các chỉ thỉ và các thông tin về quá trình này được đóng gói trong bao bì của thông điệp.

Định nghĩa giao tiếp

Yêu cầu HTTP

Yêu cầu HTTP bao gồm:

- Phương tiện yêu cầu (Request Line):
    - Phương thức gửi bắt buộc là POST
    - Xác định tài nguyên của yêu cầu cần phải đầy đủ theo URI (RFC1630)
    - Phiên bản yêu cầu cần HTTP/1.1
- Phần đầu của yêu cầu (Request Header):
    - Tham số bắt buộc Content-Type: phải là multipart/mixed và cần phải có tham số giới hạn cho tham số và nằm trong dấu “”, theo mô tả RFC 2822.

- Tham số bắt buộc máy chủ Host: Tên của hostname hoặc hostname:portnumber.
- Tham số bắt buộc Cache-Control: cần phải đưa ra giá trị no-cache
- Tham số bắt buộc MIME-Version: giá trị phiên bản 1.0
- Tham số bắt buộc Content-Length: chứa giá trị trong nội dung yêu cầu

- Thân của yêu cầu (Request Body):

- Phần thân của yêu cầu chứa nội dung của thông điệp. Thông điệp của tác tử chứa 2 thành phần theo đặc tả RFC2046 cho đa nội dung/trộn: bao bì thông điệp theo FIPA và nội dung của thông điệp theo FIPA.
- Phần mã hoá thân của thông điệp được chia thành hai phần, phần đầu chứa bao bì của thông điệp theo chuẩn FIPA, phần sau chứa nội dung của FIPA sẽ được gửi. Mỗi phần của FIPA cần phải chỉ ra mức độ mã hoá Content-Type theo MIME type. Mỗi phần chứa nội dung của header như Content-Transfer-Encoding nhưng yêu cầu xử lý các tham số này không bắt buộc [37].
- Tập kí tự được sử dụng trong phần đầu và dấu phân cách giữa các phần thường là các dâu nhị phần ASCII
- Mỗi khi có thể việc mã hoá kí tự các chuỗi thông điệp FIPA thành yêu cầu HTTP cần phải chỉ ra tham số charset trong Content-type. Tham số này cũng giống như tham số của bao bì của thông điệp theo chuẩn FIPA.

Mã hoá phần thân của thông điệp cần phải theo cấu trúc sau:

- Phần đầu MIME (bao gồm phần đầu MIME-Version và Content-Type)
- Dấu phân cách dòng trống của phần đầu MIME với phần thân MIME
- Dấu phần cách dòng danh giới giữa phần đầu của bao bì
- Dòng Content-Type cần phải chứa giá trị hợp lệ "application" "/" <string>, với chuỗi là giá trị được miêu tả trong đặc tả của bao bì thông điệp.
- Dòng trống (CRLF)
- Bao bì thông điệp FIPA
- Dấu phân cách dòng danh giới giữa bao bì và nội dung của thông điệp FIPA

- Dấu phân cách dòng danh giới kết thúc của thông điệp FIPA. Dòng thông điệp này là có thể là dấu báo hiệu kết thúc dòng [38].

Hồi đáp HTTP

Một hồi đáp HTTP bao gồm:Phương tiện hồi đáp (Response Line): hiên bản của hồi đáp phải là HTTP/1.1. Mã trạng thái của hồi đáp cần phải nêu rõ mã trả về thành công hoặc thất bại theo RFC2616. Trạng thái thành công chỉ ra tác tử đã nhận được nội dung thông điệp và trích xuất nội dung của thông điệp thành công từ các yêu cầu HTTP. Các thông tin chi tiết liên quan đến HTTP để phân tích cú pháp xử lý nội dung thông điệp và có thể hồi đáp lại bằng một thông điệp phản hồi khác. Nếu trạng thái thất bại thì có thể nhận được mã lỗi và mong đợi việc gửi thông điệp sử dụng kết hợp giữa địa chỉ nguồn, kiểu nội dung hoặc dừng chuyển [37].

- Phần đầu hồi đáp (Response Headers)

+ Tham số Content-Type: có thể là kiểu MIME theo RFC2045

+ Tham số Cache-Control: cần chứa giá trị no-cache

+ Tham số bắt buộc Content-Length: chỉ ra kích thước trong phần thân của hồi đáp

+ Phần thân hồi đáp (Response Body): Phần thân của hồi đáp chứa nội dung của thông điệp hồi đáp và phụ thuộc vào kiểu nội dung có thể là text, nhị phân hoặc đa phần. Phần người gửi có thể bắt buộc để đọc và sử dụng nội dung [38].

3.6 Cải thiện mô hình CCNMA bảo mật bằng cách triển khai khóa công khai MA-PKI

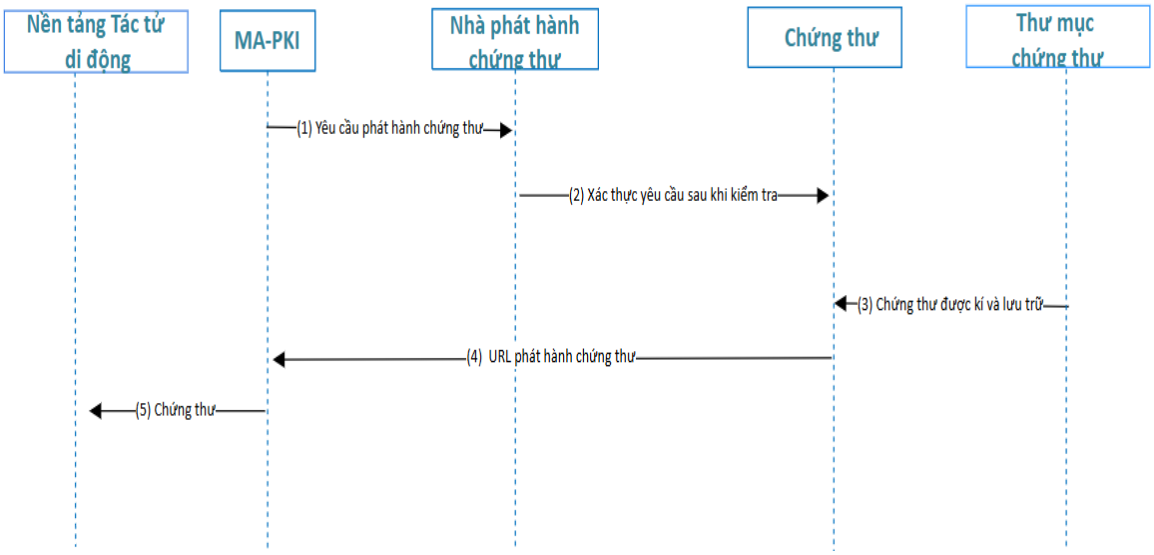
Để đảm bảo an toàn thông tin và bảo mật cho hệ thống tác tử di động, việc bổ sung PKI (Public Key Infrastructure – Hạ tầng chứng thư số) giúp cho hệ thống quản lý an toàn và bảo mật.

Do một số hạn chế của việc nhận yêu cầu, chứng thư số cho CA, chẳng hạn như nó yêu cầu nhận dạng người dùng ngoài CA để thu thập định danh và mật khẩu cho yêu cầu chứng thư, điều này không thực tế. Thiết bị nhỏ như điện thoại di động, tác tử di động tự tạo thông báo yêu cầu chứng thư, điều này có thể không phải lúc nào cũng có thể thực hiện được đối với thiết bị có ít bộ nhớ hơn và CPU kém mạnh hơn [84].

Để khắc phục những vấn đề này, việc đề xuất MA-PKI (Mobile Agent - PKI) dựa trên Mobile-PKI (M- PKI) sử dụng mật mã đường cong elip (ECC), trong đó MA-PKI thực hiện tất cả các hoạt động mật mã có lợi cho Tác tử di động. Trong sơ đồ của việc thay thế RSA, ECC được sử dụng cung cấp bảo mật RSA tương tự với thời gian xử lý tương đối ít hơn và kích thước khóa ít hơn, ví dụ: khóa 1024 bit trong RSA tương đương với 160 bit trong ECC. Hơn nữa, MA-PKI giữ toàn bộ trách nhiệm thay mặt cho tác tử di động để lưu trữ tất cả thông tin mật mã của các thiết bị nhỏ và thực hiện tất cả các hoạt động để nhận chứng chỉ từ CA [84].

Quy trình quản lý chứng thư dựa trên Tác tử di động

Quy trình quản lý chứng chỉ dựa trên MA-PKI được đề xuất được thảo luận dưới đây, trong đó sơ đồ quy trình làm việc được hiển thị trong Hình 3.1 để minh họa trực quan.



Hình 3.1: Sơ đồ quy trình làm việc của quy trình quản lý chứng thư

Mã giả về quy trình quản lý hạ tầng chứng thư MA-PKI được nêu chi tiết tại Bảng 3.5.

Bảng 3.5: Mã giả Psedocode về quy trình quản lý hạ tầng chứng thư MA-PKI

Bước 1: *MA-PKI tạo cặp khóa công khai-riêng tư dựa trên ECC cho Tác tử di động và gửi đến RA dưới dạng thông báo yêu cầu chứng chỉ cùng với các thông tin liên quan khác để lấy chứng chỉ khóa công khai của Tác tử di động từ CA.*

Bước 2: *Sau khi nhận, RA xác thực MA-PKI và xác minh hàm POP để đảm bảo rằng người dùng sở hữu khóa riêng tương ứng với khóa công khai mà chứng chỉ được yêu cầu. Nếu xác minh đạt, RA sẽ phê duyệt thông báo yêu cầu chứng chỉ và gửi nó đến CA kèm theo chứng thực để tạo chứng chỉ.*

Bước 3: *CA kiểm tra xem thông báo yêu cầu chứng chỉ có được chứng thực bởi RA hay không, nếu có, CA tạo chứng chỉ khóa công khai, ký và xuất bản chứng chỉ trong thư mục của nó.*

Bước 4: *CA tạo chứng chỉ, lưu trữ tại thư mục CA và gửi URL chứng chỉ đến MA-PKI.*

Bước 5: *Khi nhận, MA-PKI chuyển tiếp tương tự đến Tác tử di động cùng với khóa riêng của Tác tử di động được mã hóa bằng khóa bí mật được chia sẻ trước k.*

Giờ đây, Tác tử di động lấy URL chứng chỉ và giải mã khóa riêng được mã hóa bằng khóa bí mật được chia sẻ trước k và lấy khóa riêng của mình.

3.7 Tăng cường bảo mật mạng cho hệ thống mạng trên đám mây, SDN và các hệ thống mạng khác ứng dụng Tác tử di động

Tác tử di động là một công nghệ bảo mật mạng đầy hứa hẹn có thể giảm độ trễ và lưu lượng mạng, chạy trực tiếp trên máy tính đích và triển khai linh hoạt mã và dữ liệu tại đích. Do các lỗi bảo mật của kiến trúc SDN, phần này sẽ nêu bật các lĩnh vực quan trọng khác nhau, trong đó Tác tử di động là một kỹ thuật mới để tăng cường bảo mật SDN.

3.7.1 Mô hình kiến trúc tác tử di động quản lý mạng đám mây (CNMMA)

Mô hình kiến trúc CNMMA đã đề xuất việc sử dụng công nghệ Tác tử di động trong SDN và Đám mây cho mục đích quản lý mạng và giám sát an ninh.

Mô hình CNMMA bao gồm các thành phần chính sau:

- Quản lý nền tảng tác tử di động (MAP).
- Tác tử di động quản lý mạng (NMMA).
- Khung hệ thống phát hiện xâm nhập phân tán tác tử di động (MA-DIDS).

- Ứng dụng điều khiển SDN tác tử di động.

3.6.2 Quản lý nền tảng tác tử di động (MAP)

Một nền tảng tác tử được gọi là MA là cần thiết để tác tử di động trở nên sống động. Trong nghiên cứu này đã sử dụng nền tảng tác tử di động để quản lý MA và tùy chỉnh nó để tăng cường bảo mật và hiệu suất. Bằng cách lưu trữ và chạy các tác tử song song, Nền tảng MA tạo điều kiện cho tác tử truy cập vào các dịch vụ, giao tiếp với các tác tử khác và tính di động đến các Nền tảng MA thay thế. Ngoài ra, nó kiểm soát tác tử thực thi và chặn phần độc hại không mong muốn truy cập vào các tác tử [106].

3.7.3 Quản lý mạng tác tử di động

Giao thức CMIP để quản lý mạng để kiểm soát thủ công, bảo mật và lọc thông tin quản lý tốt hơn, nó đáp ứng việc quản lý mạng nhưng đòi hỏi tài nguyên và chậm hơn SNMP. Đồng thời đã khắc phục những hạn chế của giao thức SNMP và CMIP và đã phát triển một giao thức proxy để chuyển đổi các yêu cầu từ SNMP sang CMIP hoặc ngược lại bằng cách tận dụng tính năng Tác tử di động.

Kích hoạt các thủ tục quản lý mạng: CMISE gọi các hoạt động quản lý mạng theo tiêu chuẩn giao thức CMIP, chẳng hạn như M-GET, M-SET, M-ACTION, M-CREATE, M-DELETE và M-CANCEL-GET.

Mô hình bảo mật nâng cao: Để bảo vệ các hệ thống tác tử di động, Mô hình CNMMA cung cấp xác thực, tính khả dụng, trách nhiệm giải trình và không thoái thác.

Bảo mật hiệu suất bằng MA-PKI: MA-PKI (Mobile Agent - Public Key Infrastructure) được xây dựng trên nền tảng Mobile-PKI (Mobile- PKI) và sử dụng mật mã đường cong elip để tăng hiệu suất và giảm mức tiêu thụ pin (ECC). Tất cả các tác vụ mật mã cho nền tảng Tác tử di động đều được xử lý bởi MA-PKI. Trong hệ thống đề xuất, việc sử dụng ECC thay vì RSA để đạt được cùng một mức độ bảo mật với chi phí tính toán ít hơn nhiều và các khóa nhỏ hơn[84].

3.7.4 Đề xuất mô hình Khung hệ thống phát hiện xâm nhập phân tán tác tử di động (MA-DIDS)

Một mô hình mới cho các hệ thống phát hiện xâm nhập là việc triển khai MA

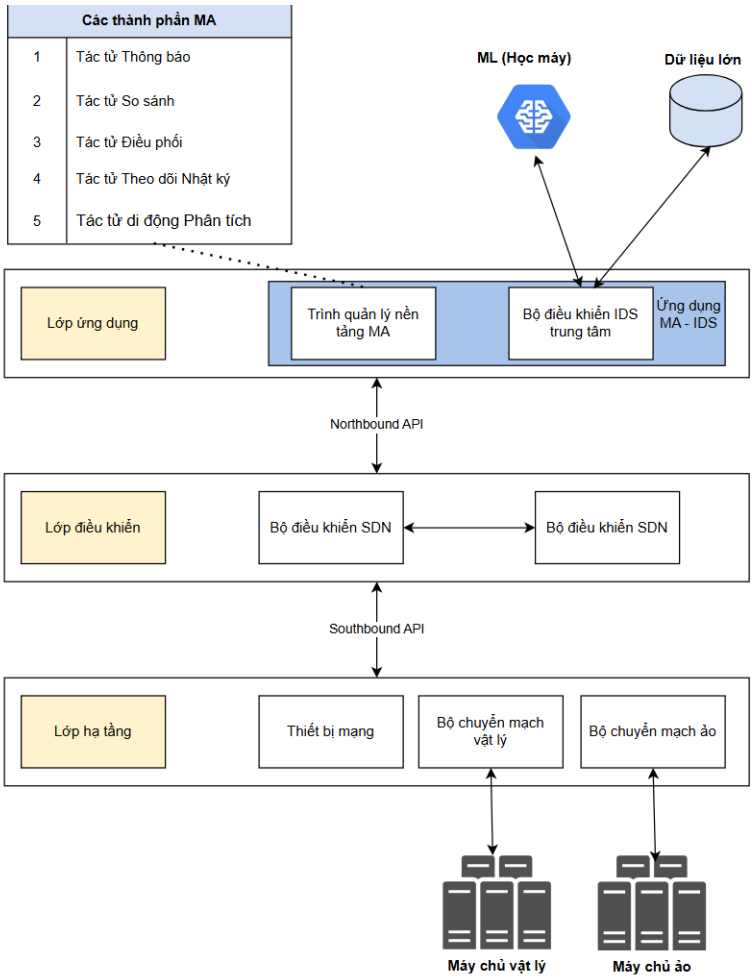
trong các hệ thống mạng SDN. Tác tử di độngs hứa hẹn cách tiếp cận mới cho hệ thống phát hiện xâm nhập so với các hệ thống hiện có vì [22]:

Cải thiện hiệu suất: Tính di động và quyền tự chủ của MA là những lợi thế chính của nó so với mô hình máy khách / máy chủ về hiệu suất.

Cải tiến thiết kế: Một mô hình mới để xác định một cuộc tấn công bảo mật được thực hiện bởi công nghệ MA.

Thời gian đáp ứng: Phản ứng tự động nhanh hơn và hiệu quả hơn đối với những kẻ tấn công được thực hiện bằng công nghệ MA.

IDS - CC và MAP Manager tạo thành hai phần chính của Khung hệ thống phát hiện xâm nhập phân tán tác tử di động (MA-DIDS) được đề xuất, như được mô tả trong hình dưới đây.



Hình 3.2: Khung hệ thống phát hiện xâm nhập phân tán tác tử di động (MA-DIDS)

Tổng quan kiến trúc khung hệ thống phát hiện xâm nhập

Kiến trúc hệ thống gồm 3 lớp trên nền tảng kiến trúc SDN tiêu chuẩn gồm 3 lớp riêng biệt:

- Lớp hạ tầng: là lớp vật lý và ảo hóa, bao gồm các máy chủ vật lý, máy chủ ảo, cùng với các thiết bị mạng như thiết bị chuyển mạch vật lý và ảo hoá. Lớp này là nền tảng thực thi các yêu cầu từ lớp trên.

- Lớp điều khiển: là bộ não của mạng, chứa thành phần trọng tâm là Bộ điều khiển SDN (SDN Controller), giao tiếp với lớp hạ tầng thông qua giao diện lập trình ứng dụng hướng nam (Southbound API) để cấu hình và điều khiển luồng dữ liệu mạng. Đồng thời, nó cung cấp giao diện hướng bắc (Northbound API) để lớp ứng dụng có thể ra quyết định và gửi yêu cầu xuống.

- Lớp ứng dụng: là lớp cao nhất, nơi chứa logic nghiệp vụ và các ứng dụng thông minh. Đối với kiến trúc này, đây chính là nơi hệ thống MA-IDS hoạt động.

Phân tích chi tiết các thành phần chính

Hệ thống Tác tử di động (Mobile Agent - MA)

- Đây là hạt nhân độc đáo của kiến trúc, giúp hệ thống trở nên linh hoạt và chủ động. Thay vì một hệ thống giám sát tập trung và bị động, các tác tử là những chương trình nhỏ, có khả năng tự di chuyển qua các nút mạng để thu thập thông tin và thực thi nhiệm vụ.

- MA Platform Manager (Trình quản lý Nền tảng Tác tử): Là môi trường để quản lý vòng đời của các tác tử, bao gồm việc tạo, triển khai, điều phối và hủy bỏ chúng.

Các thành phần của Tác tử Di động (MA Components)

- Log Tracking Agent (LTA - Tác tử Theo dõi Nhật ký): Nhiệm vụ chính là di chuyển đến các máy chủ ảo (VM) để thu thập nhật ký hệ thống (system logs), thông tin về các hoạt động mạng và sự kiện hệ điều hành. Sau đó, nó báo cáo dữ liệu này về Trung tâm Điều khiển IDS.

- Analysis Mobile Agent (Tác tử di động Phân tích): Tác tử này nhận dữ liệu

thô từ LTA để thực hiện phân tích sơ bộ, lọc nhiễu và tìm kiếm các dấu hiệu bất thường ban đầu.

- Comparator Agent (Tác tử So sánh): So sánh các hoạt động đáng ngờ đã được phát hiện với cơ sở dữ liệu mẫu xâm nhập đã biết. Nếu có sự trùng khớp, nó sẽ xác nhận đây là một cuộc tấn công đã biết.

- Mobile Agent Coordinator (Tác tử Điều phối): Là "chỉ huy trưởng" của các tác tử khác. Nó chịu trách nhiệm điều phối hoạt động, gửi các tác tử đến đúng vị trí cần thiết và đảm bảo luồng xử lý thông tin diễn ra một cách chính xác.

- Notify Agent (Tác tử Thông báo): Khi một cuộc tấn công được xác nhận (bởi Comparator Agent hoặc bởi mô hình Học máy), tác tử này sẽ được kích hoạt để gửi cảnh báo ngay lập tức đến quản trị viên hệ thống thông qua ứng dụng MA-IDS.

Trung tâm Điều khiển IDS

- Đây là trung tâm thần kinh của toàn bộ hệ thống an ninh, nơi hội tụ thông tin và ra quyết định.

- Chức năng: Nhận toàn bộ dữ liệu từ MA Platform Manager (do các tác tử báo cáo về) và kết quả phân tích từ khối Học máy.

- MA-IDS App: Là ứng dụng giao diện cho quản trị viên, hiển thị các cảnh báo, trạng thái hệ thống, và cho phép họ tương tác, điều tra các sự cố.

- Quản lý trạng thái: Trung tâm Điều khiển IDS chịu trách nhiệm cập nhật và duy trì trạng thái của từng máy ảo trong mạng.

Cơ sở dữ liệu & Dữ liệu lớn (Database & Big Data)

Hệ thống cần một kho dữ liệu lớn để lưu trữ và phân tích, bao gồm nhiều cơ sở dữ liệu với các chức năng riêng biệt:

- Cơ sở dữ liệu mẫu xâm nhập: Đây là cơ sở dữ liệu chứa các "chữ ký" (signatures) hoặc mẫu của các cuộc tấn công đã biết. Notify Agent và Comparator Agent sử dụng dữ liệu này để so sánh và xác định các mối đe dọa quen thuộc.

- Cơ sở dữ liệu sự kiện: Toàn bộ nhật ký hệ thống và các sự kiện do Log Tracking Agent (LTA) báo cáo về được lưu trữ tại đây dưới dạng Dữ liệu lớn (Big

Data). Đây là nguồn dữ liệu đầu vào cho khối Học máy.

- Cơ sở dữ liệu trạng thái máy ảo: Được cập nhật bởi Trung tâm điều khiển IDS, cơ sở dữ liệu này theo dõi trạng thái của mỗi máy ảo, có thể ở một trong ba trạng thái: bình thường (normal), bị xâm phạm (compromised), hoặc đang di chuyển (migrated).

Học máy & Khai thác dữ liệu (Machine Learning & Data Mining)

- Module này cung cấp trí thông minh để phát hiện các mối đe dọa mới, chưa từng được biết đến (zero-day attacks) dựa trên các hành vi bất thường.

- Cơ chế hoạt động: Mô hình học máy được áp dụng trên tập dữ liệu lớn bao gồm các cuộc tấn công đã được phát hiện trước đây, nhật ký hệ thống và dữ liệu do LTA thu thập. Từ đó, mô hình sẽ "học" được thế nào là hành vi "bình thường" của mạng. Bất kỳ hoạt động nào lệch khỏi trạng thái bình thường này sẽ bị coi là đáng ngờ.

- Đào tạo mô hình: Các nhà khoa học dữ liệu sử dụng các bộ dữ liệu công khai có sẵn để đào tạo mô hình. Một trong những bộ dữ liệu phổ biến nhất cho việc phát hiện xâm nhập dựa trên sự bất thường là NSL-KDD. Đây là phiên bản cải tiến của bộ dữ liệu KDD99, bao gồm các loại tấn công điển hình như:

+ DoS (Denial of Service): Tấn công từ chối dịch vụ.

+ Probe: Tấn công thăm dò, quét cổng.

+ U2R (User to Root): Tấn công leo thang đặc quyền từ người dùng thường lên quản trị viên.

+ R2L (Remote to Local): Tấn công truy cập từ xa vào máy cục bộ.

3.7.5. Trung tâm điều khiển IDS

Tất cả các chức năng của Máy ảo thông thường và các chức năng được liệt kê dưới đây được xử lý bởi Trung tâm điều khiển hệ thống phát hiện xâm nhập (IDS - CC), làm cho nó trở thành điểm quản trị tập trung cho tất cả các thành phần IDS:

Cơ sở dữ liệu & Dữ liệu lớn: Cần có cơ sở dữ liệu mẫu xâm nhập mà Notify Agent có thể sử dụng để kích hoạt cảnh báo nếu mẫu khớp với các hoạt động đáng ngờ đã được phát hiện. Tất cả ID sự kiện do LTA báo cáo được lưu trữ trong một cơ

sở dữ liệu riêng biệt. Trạng thái của máy ảo cũng phải được cập nhật bởi Trung tâm điều khiển IDS, có thể ở một trong ba trạng thái: bình thường, bị xâm phạm hoặc di chuyển.

Mô tả hệ thống Tác tử di động

Hệ thống Tác tử di động không phải là một công cụ, mà là một nền tảng điều khiển các tác tử hoạt động tự chủ bên trong mạng cục bộ hoặc đám mây, thay vì chỉ quan sát từ một tháp canh trung tâm.

Luồng hoạt động và tương tác của các Tác tử

Quá trình phát hiện và phản ứng một mối đe dọa diễn ra như một chuỗi nhiệm vụ phối hợp:

Bước 1: Điều phối và giao nhiệm vụ

Tác tử Điều phối đóng vai trò là Tổng chỉ huy: sẽ không tự mình đi thu thập dữ liệu mà sẽ ra quyết định chiến lược, dựa trên lịch trình định sẵn (ví dụ: "kiểm tra toàn bộ các máy chủ web mỗi giờ") hoặc một tín hiệu bất thường ban đầu, nó sẽ quyết định:

- Cần triển khai loại tác tử nào?

- Mục tiêu là máy chủ ảo (VM) nào?

- Nhiệm vụ cụ thể là gì? (ví dụ: thu thập nhật ký xác thực trong 5 phút vừa qua).

Bước 2: Thu thập dữ liệu tại hiện trường

Tác tử theo dõi nhật ký được lệnh và di chuyển qua mạng đến máy ảo mục tiêu. Tại đây, tác tử nhật ký theo dõi sẽ thu thập dữ liệu chi tiết hơn là chỉ "log" chung chung:

- Nhật ký hệ điều hành: Các lời gọi hệ thống, sự kiện đăng nhập/đăng xuất, các tiến trình đang chạy.

- Lưu lượng mạng: Tiêu đề các gói tin đi và đến máy ảo, các cổng đang mở, các kết nối đang hoạt động.

- Hoạt động File: Các file vừa được tạo, sửa đổi, hoặc xóa. Bằng cách thu thập và xử lý sơ bộ ngay tại nguồn, Tác tử nhật ký giảm đáng kể lượng dữ liệu phải gửi qua mạng về trung tâm, tránh gây tắc nghẽn.

Bước 3: Phân tích và đối chiếu

- Dữ liệu thô từ Tác theo dõi nhật ký được chuyển cho Tác tử phân tích. Tác tử này giống như một chuyên gia phân tích hiện trường, sàng lọc và định dạng dữ liệu, tìm kiếm những điểm bất thường (ví dụ: một tiến trình lạ đang chạy, một chuỗi đăng nhập thất bại liên tục).

- Các điểm bất thường này sau đó được chuyển cho Tác tử so sánh. Tác tử này truy cập vào Cơ sở dữ liệu mẫu xâm nhập (chứa chữ ký của các cuộc tấn công đã biết) và thực hiện đối chiếu.

Ví dụ cụ thể: Comparator Agent phát hiện một chuỗi ký tự ...OR 1=1;-- trong một yêu cầu gửi đến máy chủ web. Nó đối chiếu và thấy chuỗi này trùng khớp 100% với chữ ký của một cuộc tấn công "SQL Injection" trong cơ sở dữ liệu.

Bước 4: Xác nhận và cảnh báo

- Khi Tác tử so sánh tìm thấy sự trùng khớp, nó gửi tín hiệu "khẳng định mối đe dọa".

- Tác tử thông báo ngay lập tức được kích hoạt. Nó không chỉ gửi một cảnh báo chung chung, mà sẽ tạo một báo cáo chi tiết ví dụ như:

+ **Loại tấn công:** SQL Injection.

+ **Mức độ nghiêm trọng:** Cao.

+ **Nguồn tấn công:** Địa chỉ IP.

+ **Mục tiêu bị ảnh hưởng:** Máy chủ ảo Web-Server-01.

+ **Thời gian:** 11/08/2025 05:13 AM.

- Tiếp đó, báo cáo này được đẩy về Trung tâm Điều khiển để xử lý.

Học máy & Khai thác dữ liệu: Trong module này, học máy được áp dụng cho

cơ sở dữ liệu hệ thống chứa thông tin về các xâm nhập được phát hiện, nhật ký hệ thống và dữ liệu từ Tác tử theo dõi nhật ký để suy ra thông tin về các xâm nhập mới. Các nhà khoa học dữ liệu sử dụng nhiều bộ dữ liệu công khai có sẵn rộng rãi để đào tạo các mô hình học máy. Một trong những bộ dữ liệu phổ biến nhất cho các mô hình phát hiện xâm nhập dựa trên sự bất thường là NSL-KDD, là phiên bản cải tiến của bộ dữ liệu KDD99 và bao gồm các loại tấn công như: DoS, Probe, U2R và R2L...

Nếu hệ thống tác tử là những người lính tuần tra tìm kiếm kẻ thù đã biết mặt, thì module Học máy là chuyên gia phân tích, có khả năng phát hiện các mối đe dọa chưa từng có tiền lệ dựa trên hành vi.

Luồng hoạt động chi tiết của module học máy

Giai đoạn 1: Huấn luyện Mô hình

- Đây là quá trình diễn ra trong phòng thí nghiệm, không ảnh hưởng đến hoạt động của mạng.

- Tổng hợp dữ liệu: Các nhà khoa học dữ liệu lấy dữ liệu từ hai nguồn chính: Dữ liệu lịch sử từ kho dữ liệu lớn của hệ thống và các bộ dữ liệu công khai chuẩn hóa như NSL-KDD.

- Kỹ thuật đặc trưng**:** Dữ liệu thô (như địa chỉ IP, loại giao thức) được chuyển đổi thành các con số mà máy tính có thể hiểu được (vector đặc trưng). Ví dụ: Giao thức "TCP" có thể được biểu diễn bằng số 1, "UDP" bằng số 2.

- Huấn luyện: Các vector đặc trưng này được đưa vào một thuật toán học máy (ví dụ: Random Forest, Neural Network). Thuật toán sẽ "học" các mẫu phức tạp để phân biệt giữa lưu lượng "bình thường" và lưu lượng "tấn công" từ bộ dữ liệu đã được gán nhãn.

- Kết quả: Đầu ra của giai đoạn này là một file mô hình đã được huấn luyện, giống như một bộ não nhân tạo đã được đào tạo.

Giai đoạn 2: Phát hiện Thời gian thực

- Đây là quá trình diễn ra liên tục trên mạng đang hoạt động.

- Tải Mô hình: Trung tâm điều khiển tải file mô hình đã được huấn luyện và sẵn sàng sử dụng.

- Luồng dữ liệu trực tiếp: Dữ liệu mới liên tục được các Tác tử theo dõi nhật ký thu thập và gửi về.

- Dự đoán: Dữ liệu mới này cũng được chuyển đổi thành vector đặc trưng và được đưa vào mô hình. Mô hình sẽ đưa ra một dự đoán cho mỗi hoạt động mạng, thường là một "điểm số bất thường" (anomaly score) từ 0 đến 1.

- Ngưỡng quyết định**:** Hệ thống đặt ra một ngưỡng (ví dụ: 0.9). Bất kỳ hoạt động nào có điểm số vượt ngưỡng này sẽ bị coi là một mối đe dọa tiềm tàng, ngay cả khi nó không khớp với bất kỳ chữ ký nào trong cơ sở dữ liệu.

Giai đoạn 3: Phản ứng Thông minh

- Khi một sự bất thường được phát hiện với độ tin cậy cao, Trung tâm điều khiển sẽ:

- Kích hoạt Tác tử thông báo để gửi cảnh báo "Phát hiện hành vi bất thường tiềm tàng" cho quản trị viên để điều tra sâu hơn.

- Tự động phản ứng (nâng cao): Gửi một yêu cầu qua cầu bắc xuống Điều khiển SDN, ra lệnh cho nó tự động thực hiện một hành động phòng thủ, ví dụ: "Tạm thời cách ly VM Web-Server-01 khỏi phần còn lại của mạng cho đến khi có xác nhận từ quản trị viên."

Sự kết hợp giữa hệ thống Tác tử di động và Học máy tạo ra một cơ chế phòng thủ đa lớp: phát hiện các mối đe dọa đã biết một cách nhanh chóng và hiệu quả, đồng thời nhận dạng các cuộc tấn công mới, tinh vi dựa trên phân tích hành vi thông minh.

3.8 Đánh giá so sánh về hệ thống IDS sử dụng Tác tử di động và hệ thống IDS không sử dụng tác tử di động

Dựa trên thuộc tính và tính chất của Tác tử di động, hệ thống IDS dựa trên Tác tử di động đem lại ưu thế và hiệu quả đối với các hệ thống mạng trên Đám mây, IoT và các hệ thống phân tán.

Bảng 3.6: So sánh hệ thống IDS sử dụng Tác tử di động và hệ thống IDS không sử dụng Tác tử di động

TT	Tiêu chí	IDS sử dụng Tác tử di động	IDS không sử dụng Tác tử di động	Giải thích
1	Tính linh hoạt	Tác tử di động có thể di chuyển giữa các nút mạng để thu thập và phân tích dữ liệu tại chỗ.	Cố định tại một vị trí, phụ thuộc vào việc gửi dữ liệu về trung tâm để phân tích.	- IDS sử dụng Tác tử di động: Tác tử di động có thể di chuyển giữa các nút mạng để thu thập và phân tích dữ liệu tại chỗ, giúp hệ thống linh hoạt hơn trong việc phát hiện và xử lý sự cố. - IDS không sử dụng Tác tử di động: Hệ thống cố định tại một vị trí, phụ thuộc vào việc gửi dữ liệu về trung tâm để phân tích, dẫn đến độ linh hoạt thấp hơn.
2	Khả năng mở rộng	Dễ dàng mở rộng để quản lý các hệ thống mạng lớn và phân tán.	Khó mở rộng hơn do phụ thuộc vào cơ sở hạ tầng trung tâm.	- IDS sử dụng Tác tử di động: Dễ dàng mở rộng để quản lý các hệ thống mạng lớn và phân tán do Tác tử di động có thể tự động di chuyển và thích ứng với các nút mạng mới. - IDS không sử dụng Tác tử di động: Khó mở rộng hơn do phụ thuộc vào cơ sở hạ tầng trung tâm, cần nâng cấp phần cứng và phần mềm để đáp ứng nhu cầu mở rộng.

TT	Tiêu chí	IDS sử dụng Tác tử di động	IDS không sử dụng Tác tử di động	Giải thích
3	Tốc độ phản hồi	Phản hồi nhanh do xử lý dữ liệu tại chỗ, không cần truyền tải dữ liệu về trung tâm.	Chậm hơn do cần truyền tải dữ liệu về trung tâm để phân tích.	- IDS sử dụng Tác tử di động: Phản hồi nhanh do xử lý dữ liệu tại chỗ, không cần truyền tải dữ liệu về trung tâm. - IDS không sử dụng Tác tử di động: Chậm hơn do cần truyền tải dữ liệu về trung tâm để phân tích, dẫn đến độ trễ cao hơn.
4	Tải mạng	Giảm tải mạng do xử lý dữ liệu tại chỗ, chỉ gửi kết quả về trung tâm.	Tăng tải mạng do cần truyền tải toàn bộ dữ liệu về trung tâm.	- IDS sử dụng Tác tử di động: Giảm tải mạng do xử lý dữ liệu tại chỗ, chỉ gửi kết quả về trung tâm. - IDS không sử dụng Tác tử di động: Tăng tải mạng do cần truyền tải toàn bộ dữ liệu về trung tâm, gây áp lực lên băng thông mạng.
5	Tự động hóa	Tự động di chuyển và thực hiện nhiệm vụ mà không cần can thiệp thủ công.	Phụ thuộc nhiều vào cấu hình và quản lý thủ công từ trung tâm.	- IDS sử dụng Tác tử di động: Tự động di chuyển và thực hiện nhiệm vụ mà không cần can thiệp thủ công, giúp tăng hiệu quả quản lý. - IDS không sử dụng Tác tử di động: Phụ thuộc nhiều vào cấu hình và quản lý thủ công từ

TT	Tiêu chí	IDS sử dụng Tác tử di động	IDS không sử dụng Tác tử di động	Giải thích
				trung tâm, dẫn đến khả năng tự động hóa thấp hơn.
6	Khả năng phát hiện tấn công	Phát hiện tấn công dựa trên hành vi và dấu hiệu tại chỗ, hiệu quả với các tấn công phân tán.	Phát hiện tấn công dựa trên dữ liệu tổng hợp, có thể bỏ sót các tấn công cục bộ.	- IDS sử dụng Tác tử di động: Phát hiện tấn công dựa trên hành vi và dấu hiệu tại chỗ, hiệu quả với các tấn công phân tán và cục bộ. - IDS không sử dụng Tác tử di động: Phát hiện tấn công dựa trên dữ liệu tổng hợp, có thể bỏ sót các tấn công cục bộ hoặc phân tán.
7	Bảo mật	Có thể bảo mật cao hơn do xử lý dữ liệu tại chỗ, giảm nguy cơ rò rỉ thông tin.	Dễ bị tấn công vào trung tâm xử lý dữ liệu.	- IDS sử dụng Tác tử di động: Có thể bảo mật cao hơn do xử lý dữ liệu tại chỗ, giảm nguy cơ rò rỉ thông tin. - IDS không sử dụng Tác tử di động: Dễ bị tấn công vào trung tâm xử lý dữ liệu, gây rủi ro bảo mật cao hơn.
8	Chi phí triển khai	Chi phí ban đầu cao hơn do cần phát triển và triển khai Tác tử di động.	Chi phí ban đầu thấp hơn, nhưng chi phí vận hành và bảo trì có thể cao hơn.	- IDS sử dụng Tác tử di động: Chi phí ban đầu cao hơn do cần phát triển và triển khai Tác tử di động, nhưng chi phí vận hành và bảo trì thấp hơn. - IDS không sử dụng Tác tử di

TT	Tiêu chí	IDS sử dụng Tác tử di động	IDS không sử dụng Tác tử di động	Giải thích
				động: Chi phí ban đầu thấp hơn, nhưng chi phí vận hành và bảo trì có thể cao hơn do phụ thuộc vào cơ sở hạ tầng trung tâm.
9	Khả năng thích ứng	Dễ dàng thích ứng với các môi trường mạng động và phức tạp.	Khó thích ứng với các môi trường mạng động do cấu trúc tập trung.	- IDS sử dụng Tác tử di động: Dễ dàng thích ứng với các môi trường mạng động và phức tạp, như hệ thống Đám mây hoặc IoT. - IDS không sử dụng Tác tử di động: Khó thích ứng với các môi trường mạng động do cấu trúc tập trung và phụ thuộc vào trung tâm.
10	Triển khai ứng dụng	Phù hợp với hệ thống mạng Đám mây, IoT, và các hệ thống phân tán.	Phù hợp với hệ thống mạng nhỏ và tập trung.	- IDS sử dụng Tác tử di động: Phù hợp với hệ thống mạng Đám mây, IoT, và các hệ thống phân tán. - IDS không sử dụng Tác tử di động: Phù hợp với hệ thống mạng nhỏ và tập trung.

3.9 Đánh giá thử nghiệm so sánh hiệu suất giữa mã hoá truyền tin nền tảng Tác tử di động sử dụng thuật toán ECC và RSA

3..1 Bộ dữ liệu thử nghiệm

Dữ liệu thời gian (đơn vị: milliseconds) được thu thập từ 20 lần chạy cho mỗi cài đặt thuật toán ECC và RSA và số lượng gửi 1000 bản tin:

Bảng 3.7: Bảng số liệu số gửi bản tin lượt chạy tương ứng sử dụng thuật toán ECC và RSA

Lần chạy	ECC (256-bit)	RSA (3072-bit)
1	142	298
2	138	305
3	145	287
4	140	310
5	143	295
6	137	292
7	141	301
8	139	289
9	144	303
10	142	297
11	140	308
12	139	290
13	146	306
14	138	293
15	143	299
16	137	288
17	141	296
18	144	302
19	139	291
20	142	300

Ghi chú:

- Thời gian có thể dao động nhẹ do tải CPU, mạng, hoặc JVM.

3.9.2. Phân tích kết quả thử nghiệm

Thực nghiệm 1: Mô hình mã hóa ECC (Elliptic Curve Cryptography)

Mục tiêu

Đánh giá hiệu suất mã hóa của thuật toán ECC 256-bit trong quá trình truyền thông tin giữa các tác tử di động (Mobile Agent), nhằm đo lường thời gian mã hóa trung bình và độ ổn định khi tăng số lần thử nghiệm lên 20 lần.

Cách cài đặt

- Nền tảng thử nghiệm: Nền tảng Tác tử di động chạy trên Windows 10.
- Cấu hình phần cứng: CPU Intel Core i7, RAM 32 GB.
- Mỗi lần thử nghiệm gửi 1000 bản tin được mã hóa và giải mã bằng ECC 256-bit (secp256r1).
- Ngôn ngữ: Java 21.
- Số lần chạy: 20 lần để lấy giá trị trung bình và độ lệch chuẩn.

Phân tích kết quả thu được

Thời gian trung bình của ECC qua 20 lần thử là 140,95 ms với độ lệch chuẩn 2,73 ms. Kết quả cho thấy thuật toán ECC có độ ổn định cao và tốc độ xử lý nhanh gấp khoảng 2,12 lần so với RSA. Do khóa ECC ngắn hơn nhưng vẫn đảm bảo mức độ bảo mật tương đương, nên chi phí tính toán và băng thông giảm đáng kể.

Kết luận

Thuật toán ECC thể hiện hiệu năng vượt trội về tốc độ và độ ổn định, phù hợp với các hệ thống Tác tử di động yêu cầu thời gian xử lý thấp, băng thông hạn chế và khả năng truyền tin nhanh. ECC là lựa chọn tối ưu cho môi trường đám mây và IoT.

Thực nghiệm 2: Mô hình mã hóa RSA (Rivest–Shamir–Adleman)

Mục tiêu

Đánh giá hiệu suất mã hóa của thuật toán RSA 3072-bit trong cùng điều kiện môi trường với ECC, nhằm so sánh trực tiếp về thời gian xử lý và mức độ ổn định khi thực hiện 20 lần thử nghiệm.

Cách cài đặt

- Nền tảng thử nghiệm và cấu hình phần cứng giống mô hình ECC.
- Sử dụng RSA 3072-bit với cơ chế padding PKCS#1 v1.5.
- Mỗi lần chạy mã hóa và giải mã 1000 bản tin.
- Số lần chạy: 20 lần để đảm bảo độ tin cậy thống kê.

Số liệu tổng hợp

Thời gian trung bình của RSA qua 20 lần thử là 298,55 ms với độ lệch chuẩn 6,78 ms. RSA có xu hướng dao động cao hơn ECC và mất nhiều thời gian hơn do phép toán mô-đun lớn 3072-bit.

Phân tích

RSA có tốc độ xử lý chậm hơn ECC khoảng 2,12 lần. Độ biến thiên thời gian cao hơn cho thấy RSA chịu ảnh hưởng lớn từ tải CPU và bộ nhớ. Với chi phí tính toán cao, RSA không thích hợp cho các tác tử di động thường xuyên di chuyển hoặc giao tiếp với nhiều nút mạng.

Kết luận

RSA phù hợp với môi trường yêu cầu bảo mật tuyệt đối như hệ thống chứng thực khóa công khai (PKI). Tuy nhiên, trong môi trường tác tử di động cần tốc độ và tính linh hoạt cao, ECC cho hiệu quả tốt hơn rõ rệt.

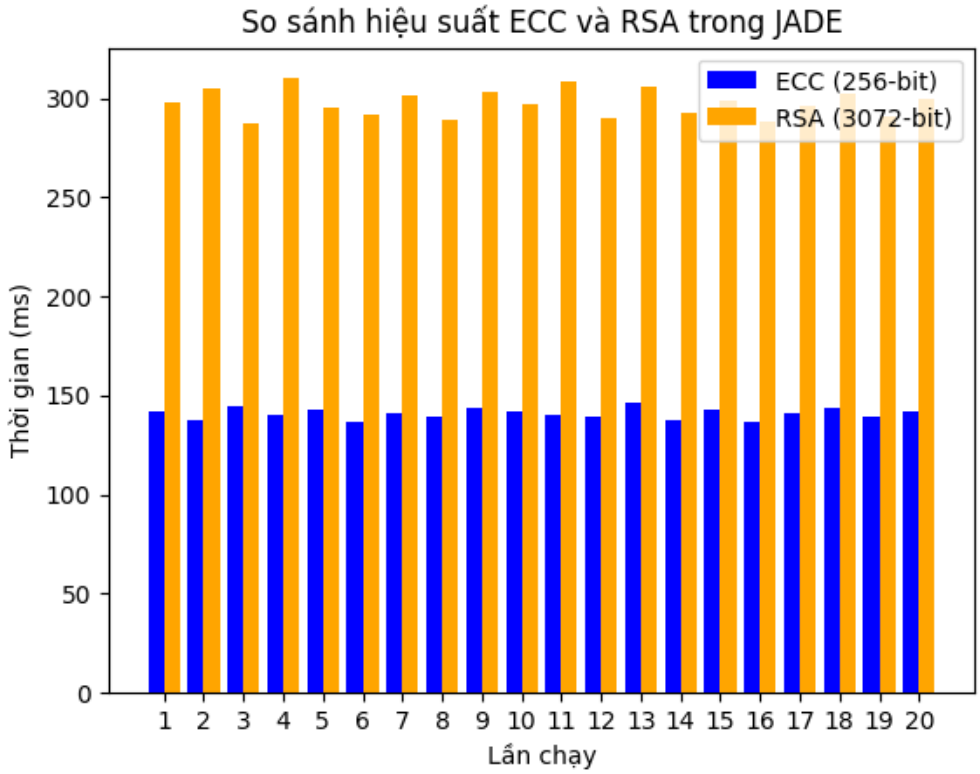
3.9.3 So sánh tổng hợp

Bảng 3.8: Bảng số liệu số gửi bản tin lượt chạy tương ứng sử dụng thuật toán ECC và RSA

Thuật toán	Trung bình (ms)	Độ lệch chuẩn (ms)	Tốc độ tương đối	Kết luận
ECC (256-bit)	140,95	2,73	Nhanh hơn 2,12	Tối ưu cho Tác tử di động
RSA (3072-bit)	298,55	6,78	Chậm hơn	Phù hợp cho bảo mật cao

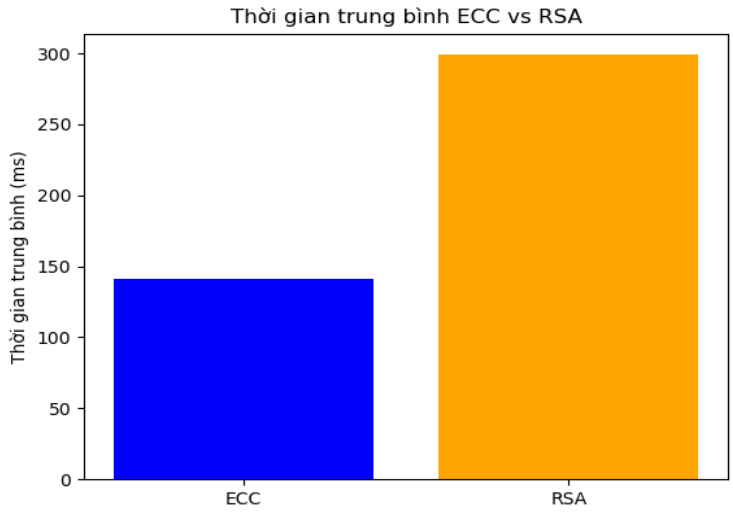
Biểu đồ so sánh hiệu suất giữa mã hoá truyền tin nền tảng Tác tử di động sử dụng thuật toán ECC và RSA

Biểu đồ so sánh hiệu suất mã hoá truyền tin nền tảng Tác tử di động sử dụng thuật toán ECC và RSA như Hình 3.3 và Hình 3.4.



Hình 3.3: Biểu đồ so sánh hiệu suất ECC và RSA trong môi trường Tác tử di động

- Thời gian so sánh trung bình ECC và RSA



Hình 3.4: Thời gian trung bình giữa ECC và RSA

So sánh hiệu suất:

- **Tỷ lệ:** $RSA/ECC = \frac{\text{RSA}}{\text{ECC}} = \frac{298,55}{140,95} \approx 2,12$

- ECC nhanh hơn RSA khoảng 2,12 lần trong thử nghiệm.

3.9.4 Kết luận đánh giá hiệu suất

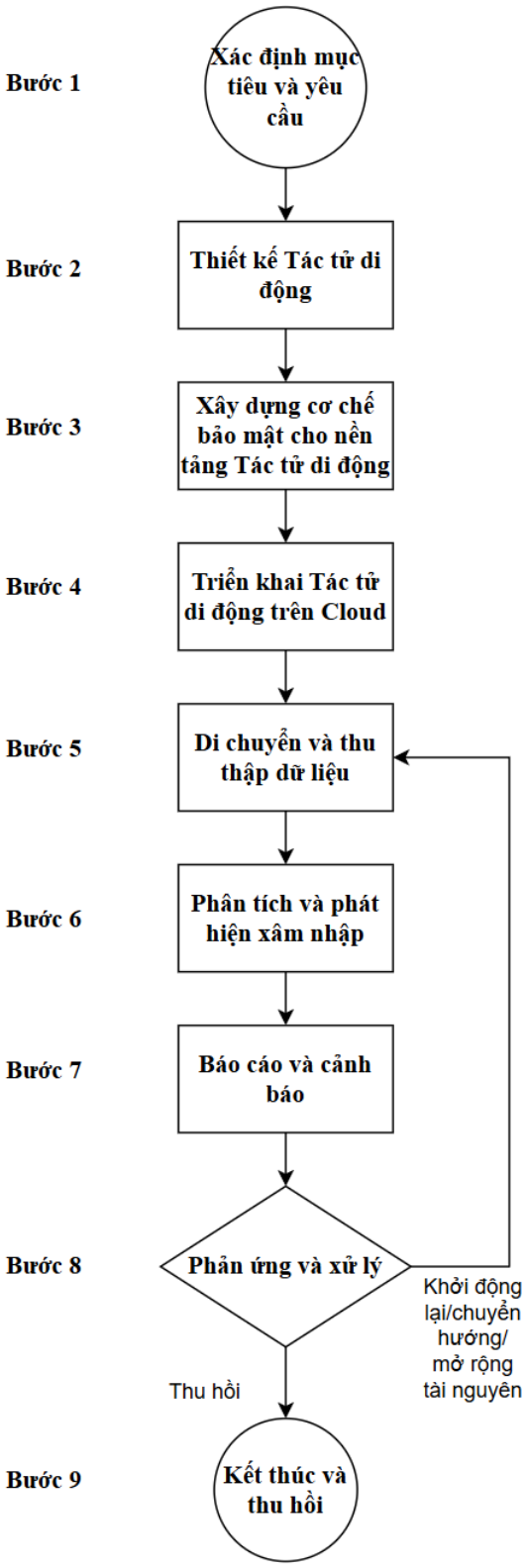
Qua 20 lần thử nghiệm, ECC thể hiện hiệu năng vượt trội về tốc độ và ổn định so với RSA. RSA chỉ thích hợp trong các ứng dụng yêu cầu mã hóa mạnh với ít giao tiếp. Trong khi đó, ECC phù hợp cho môi trường phân tán, nơi hiệu suất và băng thông là yếu tố quan trọng.

3.10 Ứng dụng mô hình Khung hệ thống phát hiện xâm nhập phân tán tác tử di động MA-DIDS

Tác tử di động là một công nghệ linh hoạt và hiệu quả để phát hiện sự xâm nhập mạng (Intrusion Detection). Tác tử di động có khả năng di chuyển giữa các nút mạng, thu thập dữ liệu, phân tích và phát hiện các hoạt động đáng ngờ hoặc tấn công mạng, nên việc xây dựng và triển khai hệ thống phát hiện xâm nhập.

3.10.1 Xây dựng quy trình triển khai hệ thống phát hiện xâm nhập phân tán tác tử di động MA-DIDS

Dưới đây là quy trình chi tiết sử dụng Tác tử di động trong phát hiện sự xâm nhập mạng:



Hình 3.5: Quy trình triển khai hệ thống phát hiện xâm nhập phân tán tác tử di động MA-DIDS

Trong đó, luồng thực hiện bao gồm các bước được thực hiện cụ thể như sau:

Bước 1. Xác định mục tiêu và yêu cầu

- Mục tiêu: Phát hiện các hoạt động xâm nhập mạng như tấn công DDoS, quét cổng, hoặc truy cập trái phép.

- Yêu cầu: Xác định các yêu cầu cụ thể như độ chính xác, thời gian phản hồi, bảo mật và khả năng mở rộng.

Bước 2. Thiết kế Tác tử di động

- Chức năng: Thiết kế các chức năng của Tác tử di động, bao gồm:

- Thu thập dữ liệu từ các nút mạng (log, lưu lượng mạng, trạng thái thiết bị).

- Phân tích dữ liệu để phát hiện các hoạt động đáng ngờ.

- Di chuyển giữa các nút mạng để thực hiện nhiệm vụ.

- Giao thức: Xác định giao thức truyền thông và cơ chế di chuyển của Tác tử di động (ví dụ: sử dụng TCP/IP hoặc các giao thức đặc biệt).

- Bảo mật: Thiết kế cơ chế bảo mật để đảm bảo Tác tử di động không bị tấn công hoặc giả mạo trong quá trình di chuyển.

Bước 3. Xây dựng cơ chế bảo mật nền tảng tác tử di động

- *Quản lý khóa cho tác tử di động*

+ Khóa cần quản lý: Khóa mã tác tử là khoá riêng (private) của quản trị viên; khóa định danh tác tử (mỗi tác tử có cặp khóa riêng khi sinh); khóa phiên (cho mỗi giao tiếp tác tử – nền tảng).

+ Thực hành triển khai: sinh khóa trên thiết bị đáng tin cậy như TrustZone, hoặc secure element (nếu có); không nhúng private key vào mã tác tử dưới dạng thuần văn bản; dùng key wrapping khi lưu trữ khóa tạm thời trong bộ nhớ ngoài.

+ Xoay khóa định kỳ: khóa phiên mỗi phiên làm việc; khóa định danh tác tử mỗi 30–90 ngày nếu tác tử hoạt động dài hạn.

- *Quản lý chứng thư khi triển khai*

+ Luồng chứng thực điển hình:

Khi nền tảng nhận tác tử: Tác tử gửi mã + chữ ký số (signature) của mã bằng private key của quản trị viên; Nền tảng kiểm tra chữ ký với public key của nhà phát

triển (được cấu hình sẵn hoặc từ Hạ tầng chứng thư); Nếu đúng mã không bị sửa, nguồn gốc đáng tin.

Khi tác tử kết nối lại nền tảng (sau khi triển khai): Dùng mutual TLS (mTLS): Client (tác tử) có private key riêng, chứng chỉ được cấp bởi CA nội bộ của nền tảng; Server (nền tảng) cũng có chứng chỉ; hoặc dùng JSON Web Token (JWT) có chữ ký kèm nonce và timestamp để chống replay.

- *Chống phát lại (replay)*

Mỗi yêu cầu (request) thêm nonce (do máy chủ cấp hoặc kết hợp timestamp + sequence number); máy chủ lưu nonce đã dùng trong một khoảng thời gian (ví dụ: 5 phút) để từ chối nonce trùng.

- *Chống tấn công xen giữa MITM*

Bắt buộc sử dụng TLS 1.3 trở lên; Xác thực hostname + certificate chain; Nếu dùng mạng không tin cậy, có thể thêm mã pin public key của máy chủ trong tác tử.

Bước 4. Triển khai Tác tử di động trên Đám mây

- Tạo Tác tử: Tạo các Tác tử di động với các chức năng cụ thể và triển khai chúng trên các nút mạng hoặc máy chủ quản lý.

- Cấu hình: Cấu hình các thông số như địa chỉ IP, cổng kết nối, và quyền truy cập cho Tác tử di động.

- Khởi chạy: Khởi chạy Tác tử di động từ một máy chủ trung tâm hoặc từ một nút mạng cụ thể.

Bước 5. Di chuyển và thu thập dữ liệu

- Di chuyển: Tác tử di động tự động di chuyển giữa các nút mạng để thu thập dữ liệu liên quan đến an ninh mạng.

- Thu thập dữ liệu: Tại mỗi nút, Tác tử di động thu thập dữ liệu như:

- Log hệ thống.

- Lưu lượng mạng (packet capture).

- Trạng thái thiết bị (CPU, bộ nhớ, kết nối mạng).

Bước 6. Phân tích và phát hiện xâm nhập

- Phân tích dữ liệu: Tác tử di động phân tích dữ liệu thu thập được để phát hiện các hoạt động đáng ngờ, chẳng hạn:

- Tấn công DDoS.

- Quét cổng (port scanning).

- Truy cập trái phép.

- Phát tán mã độc.

- Sử dụng mẫu (signature) và hành vi (behavior): Tác tử di động có thể sử dụng các mẫu tấn công đã biết (signature-based) hoặc phân tích hành vi bất thường (anomaly-based) để phát hiện xâm nhập.

Bước 7. Báo cáo và cảnh báo

- Báo cáo: Tác tử di động gửi báo cáo về máy chủ quản lý trung tâm hoặc hiển thị kết quả trực tiếp trên giao diện quản lý.

- Cảnh báo: Nếu phát hiện sự xâm nhập, Tác tử di động sẽ gửi cảnh báo đến quản trị viên hoặc kích hoạt các biện pháp phòng ngừa tự động.

Bước 8. Phản ứng và xử lý

- Phản ứng tự động: Tác tử di động có thể tự động thực hiện các hành động như:

- Chặn địa chỉ IP đáng ngờ.

- Đóng cổng dịch vụ bị tấn công.

- Khởi động lại dịch vụ hoặc thiết bị.

- Can thiệp thủ công: Quản trị viên có thể can thiệp thủ công dựa trên thông tin từ Tác tử di động.

Bước 9. Kết thúc và thu hồi

- Kết thúc nhiệm vụ: Khi hoàn thành nhiệm vụ, Tác tử di động sẽ kết thúc hoạt động và trả về kết quả cuối cùng.

- Thu hồi Tác tử: Tác tử di động có thể tự động thu hồi hoặc bị hủy bỏ nếu không cần thiết.

3.10.2 Lợi ích của việc sử dụng hệ thống phát hiện xâm nhập phân tán tác tử di động MA-DIDS trong phát hiện xâm nhập mạng

1. Tính linh hoạt: Tác tử di động có thể di chuyển giữa các nút mạng để thu thập và phân tích dữ liệu một cách linh hoạt.

2. Giảm tải mạng: Tác tử di động xử lý dữ liệu tại chỗ, giảm lượng dữ liệu cần truyền tải qua mạng.

3. Tự động hóa: Tác tử di động giúp tự động hóa quá trình phát hiện và phản ứng với các cuộc tấn công mạng.

4. Khả năng mở rộng: Tác tử di động có thể dễ dàng mở rộng để quản lý các hệ thống mạng lớn và phức tạp.

3.10.3 Ứng dụng hệ thống phát hiện xâm nhập phân tán tác tử di động MA-DIDS

- *Phát hiện tấn công DDoS*: Tác tử di động di chuyển giữa các bộ định tuyến và bộ chuyển mạch để phát hiện lưu lượng bất thường.

- *Phát hiện quét cổng*: Tác tử di động phân tích log từ các máy chủ để phát hiện các yêu cầu quét cổng đáng ngờ.

- *Phát hiện mã độc*: Tác tử di động thu thập dữ liệu từ các máy tính trong mạng để phát hiện các tệp tin hoặc tiến trình đáng ngờ.

3.11 Đánh giá thử nghiệm so sánh giữa Baseline IDS và MA-DIDS

3.11.1 Môi trường thử nghiệm

Môi trường thử nghiệm Baseline IDS và MA- DIDS được cấu hình như sau:

- CPU: 2 CPU Intel Xeon 3963 v3
- RAM: 128 GB
- JVM: Java version 21.0.9
- OS: Window Server 2019
- Network LAN : 1 Gbps
- Thử nghiệm các kỹ thuật tấn công: quét cổng, SYN flood/DDoS nhỏ, brute-force SSH, quét ARP/ARP spoof.

3.11.2 Xây dựng chuẩn hoá baseline IDS và MA-DIDS trong thực nghiệm

Mục tiêu

- Thống nhất một phương thức thực nghiệm.
- Các độ đoTP, FP, FN, TN suy ra từ thực nghiệm.
- Tách rõ tham số dùng chung và tham số chỉ mô tả lợi thế của MA-DIDS

Cách lấy số liệu

+ Luồng ML

- Lấy mẫu nhãn thật theo ML_STREAM_ATTACK_PRIOR.

- Sinh vector FEATURE_VECTOR_DIMENSION chiều qua SyntheticEvent Generator.

- Cảnh báo nếu anomaly hoặc max độ tin cậy KDD > τ (KDD_CONFIDENCE _THRESHOLD).

- Chỉ MA-DIDS: có thể đưa ra cảnh báo với MADIDS_ML_CORRELATION _RESCUE_PROB khi nhãn thật là tấn công nhưng chưa cảnh báo.

+ Luồng log

- Mỗi poll: xác suất LOG_SUSPICIOUS_EVENT_PROBABILITY có sự kiện (cùng hai hệ).

- Nhãn tấn công với LOG_EVENT_ATTACK_PRIOR (có điều kiện).

- Baseline dùng BASELINE_LOG_*; MA-DIDS dùng MADIDS_LOG_*.

Xây dựng bảng giá trị tham số đầy đủ cho cả 2 nền tảng Baseline IDS và MA-DIDS

Bảng 3.9: Bảng giá tham số chung cho Baseline IDS và MA-DIDS

Tham số	Giá trị	Ý nghĩa
KDD_CONFIDENCE_THRESHOLD	0,7	Ngưỡng KDD; hai hệ phải giống nhau.
FEATURE_VECTOR_DIMENSION	15	Chiều vector đặc trưng.
ML_STREAM_ATTACK_PRIOR	0,35	Tiền tấn công cho luồng ML.
LOG_SUSPICIOUS_EVENT_PROBABILITY	0,30	Xác suất có sự kiện log mỗi Poll.
LOG_EVENT_ATTACK_PRIOR	0,45	P(tấn công | đã có sự kiện log).
DEFAULT_MASTER_SEED	Giá trị mặc định ngẫu nhiên	Hạt giống chính + runIndex → tái lập.

Bảng 3.10: Bảng giá trị tham số cho Baseline IDS

Tham số	Giá trị	Ý nghĩa
BASELINE_LOG_MISS _GIVEN_ATTACK	0,15	P(không cảnh báo | tấn công)
BASELINE_LOG_ FPR_GIVEN_NORMAL	0,05	P(cảnh báo | bình thường)

Bảng 3.11: Bảng giá trị tham số cho MA-DIDS

Tham số	Giá trị	Ý nghĩa
MADIDS_ML_CORRELA TION _RESCUE_PROB	0,12	Cứu cảnh báo sau khi fusion ML bỏ sót (nhãn thật = tấn công).
MADIDS_LOG_MISS _GIVEN_ATTACK	0,08	Ít bỏ sót hơn baseline trên log.
MADIDS_LOG_FPR _GIVEN_NORMAL	0,03	Ít báo sai hơn baseline trên log.

3.11.3 Các bộ thực nghiệm đánh giá so sánh

- ***Thực nghiệm 1***: Đánh giá độ chính xác phát hiện (True Positive Rate & False Positive Rate).

- Mục tiêu: Xác định khả năng phát hiện chính xác và tỉ lệ cảnh báo sai của hai hệ thống Baseline IDS và MA-DIDS.

- Cách cài đặt: Mỗi hệ thống được chạy 30 lần, thời gian mỗi lần 5 giây. Dữ liệu tấn công và bình thường được trộn ngẫu nhiên để đảm bảo độ tin cậy 95%. Các chỉ số thu thập: True Positive Rate (TPR), False Positive Rate (FPR).

Bảng 3.12: Bảng số liệu so sánh đánh giá độ chính xác phát hiện tấn công xâm nhập giữa Baseline IDS và MA-DIDS

Chỉ số	Baseline	MA-DIDS	Cải thiện
True Positive Rate	99,2%	99,3%	0,2%
False Positive Rate	41,7%	15,6%	-62,7%

- Phân tích: MA-DIDS gần như giữ nguyên TPR nhưng giảm mạnh FPR tới 62,7%. Điều này chứng tỏ hệ thống MA-DIDS có khả năng lọc nhiễu tốt hơn, hạn chế cảnh báo sai.

- Kết luận: MA-DIDS có hiệu quả vượt trội về độ chính xác, đặc biệt trong việc giảm cảnh báo sai so với Baseline.

Thực nghiệm 2: Đánh giá độ tin cậy phân loại (Precision & F1-Score)

- Mục tiêu: So sánh khả năng phân loại đúng và cân bằng giữa độ chính xác và độ bao phủ của hai mô hình.

- Cách cài đặt: Sử dụng cùng tập dữ liệu kiểm thử với 30 mẫu. Đo các chỉ số: Precision, F1-Score (trung bình 30 lần).

Bảng 3.13: Bảng số liệu so sánh đánh giá độ tin cậy phân loại giữa Baseline IDS và MA-DIDS

Chỉ số	Baseline	MA-DIDS	Cải thiện
Precision	85,4%	95,6%	11,8%
F1-Score	90,7%	97,1%	7,0%

- Phân tích: MA-DIDS cải thiện Precision đáng kể (+11,8%), cho thấy ít cảnh báo sai hơn. F1-Score tăng 7% thể hiện sự cân bằng giữa độ chính xác và độ bao phủ.

- Kết luận: Hệ thống MA-DIDS đáng tin cậy hơn trong việc phân loại đúng các tấn công và giảm lỗi cảnh báo.

Thực nghiệm 3: Đánh giá hiệu suất hệ thống (Thời gian, CPU, Memory, Thread)

- Mục tiêu: So sánh mức tiêu thụ tài nguyên và tốc độ phản hồi của hai hệ thống.

- Cách cài đặt: Cùng cấu hình phần cứng, mỗi test chạy 5 giây × 30 lần. Đo các chỉ số: thời gian phát hiện, sử dụng CPU, Memory, và Thread count.

Bảng 3.14: Bảng số liệu so sánh đánh giá hiệu suất giữa Baseline IDS và MA-DIDS

Chỉ số	Baseline	MA-DIDS	Chênh lệch
Detection Time	16,6 ms	40,8 ms	+146,3%
Memory Usage	28,0%	38,5%	+37,4%
CPU Usage	46,9%	92,5%	+97,1%
Thread Count	5,5	27,9	+411,6%

- Phân tích: Mô hình MA-DIDS tiêu tốn nhiều tài nguyên hơn do cơ chế tác tử di động phân tán (do overhead của nền tảng). Thời gian phát hiện chậm hơn do nhiều bước xử lý hợp tác giữa các tác tử.

- Kết luận: Mô hình MA-DIDS phù hợp cho môi trường có tài nguyên mạnh, và đòi hỏi độ chính xác cao và phận tích sâu trong khi Baseline IDS vẫn đáp ứng cho hệ thống giới hạn tài nguyên.

3.11.4 Phân tích đánh đổi (trade-off) tài nguyên theo mô hình MA-DIDS

Mục tiêu đánh đổi tài nguyên theo mô hình MA-DIDS

- Lợi ích hướng thiết kế theo MA-DIDS: tính toán song song, tương quan đa nguồn, poll phát hiện/log có thể nhiều hơn (trong thời gian: 1,5s / 2s so 2s / 3s).

- Chi phí: Tốn tài nguyênCPU, RAM, trễ điều phối; tổng tải suy luận ML lớn hơnnếu không batch/lọc.

Bảng 3.15: Bảng phân tích đánh đổi tài nguyên theo mô hình MA-DIDS

Khía cạnh	Xu hướng điển hình (MA-DIDS so Baseline)	Giải thích ngắn
CPU	Cao hơn	Nhiều luồng tác tử, điều phối; trong mô phỏng còn cộng overhead CPU.
RAM	Cao hơn	Nhiều ngữ cảnh luồng + overhead bộ nhớ được cộng vào chỉ số.
Số thread	Cao hơn	9 luồng agent + giám sát + phát hiện + log.
Detection time	Có thể cao hơn mỗi lần gọi	Trễ điều phối (sleep) quanh bước ML; đổi lại tần suất lặp có thể cao hơn.

Phân tích theo từng chỉ số

- Theo chỉ số CPU: Đa luồng, đánh thức định kỳ, overhead điều phối; triển khai thật thêm serialize, hàng đợi, mã hóa, có thể cải thiện độ trễ nếu phân tải đúng và tổng CPU thường tăng nếu không tối ưu xử lý theo batch/lọc.

- Theo chỉ số RAM: Stack luồng, buffer, hàng đợi sự kiện, cache tương quan. Nhưng bù lại thì yêu cầu cache lớn và ít đọc ghi (I/O) lặp nhưng lưu vết (footprint) tăng.

- Theo chỉ số Threads: Mô hình một agent tương đương một thread đơn giản nhưng dễ gây thay đổi ngữ cảnh (context switch); giải pháp sử dụng thread pool (bể thread) để gộp vai trò.

- Theo chỉ số Detection time: Thành phần: suy luận ML (Machine Learning) , sleep điều phối và đọc ghi (I/O) log; giúp tăng tần suất poll có thể giảm thời gian chờ đến lần quét kế tiếp nhưng tăng tổng số lần gọi mô hình.

Đề xuất biện pháp giảm overhead

- Kiến trúc luồng: Thay đổi kiến trúc luồng (thread) xử lý theo đề xuất như trong Bảng 3.15.

Bảng 3.16: Bảng đề xuất biện pháp giảm tải tài nguyên theo mô hình MA-DIDS

Biện pháp	Mô tả	Tác động dự kiến
Gộp Tác tử trên luồng thực thi (executor thread)	ExecutorService pool 2–4 worker + hàng đợi ưu tiên thay vì 9 thread riêng	Giảm thread OS, giảm context switch
Reactive / async I/O	Một vòng lặp sự kiện thay vì sleep luân phiên	Ít "đánh thức" vô ích
Đường nóng / đường lạnh	Chỉ dùng tác tử kích thước nhỏ trên đường dữ liệu; huấn luyện / báo cáo lịch thấp	Giảm CPU trung bình

- Suy luận ML:

+ Một instance model dùng chung — tránh nhân đôi trọng số; hàng đợi inference đơn luồng nếu cần.

+ Batch inference khi model hỗ trợ.

+ Model nhẹ / lượng tử hóa — giảm FLOPs.

+ Lọc rule-based trước ML.

- Bộ nhớ & dữ liệu:

+ ArrayBlockingQueue có giới hạn; chính sách drop khi quá tải.

+ Tái sử dụng buffer trên đường nóng.

+ Nén snapshot log nếu phải giữ lịch sử tương quan.

- Mạng & điều phối (triển khai thật)

+ Giao thức nhị phân nhẹ; heartbeat thấp.

+ Phân vùng theo node để tránh trùng việc.

- Máy chủ ảo Java (JVM) & vận hành:

+ Tăng tài nguyên RAM cho máy ảo sử dụng lệnh -Xmx / -Xms cân bằng GC.

+GC low-latency (ZGC, Shenandoah) khi có độ trễ SLA.

Kết luận về đánh đổi tài nguyên

MA-DIDS trong mô hình hiện tại thể hiện chi phí tài nguyên CPU/RAM/thread cao hơn Baseline IDS để đại diện điều phối đa tác tử và tần suất xử lý khác nhau. Để

giải quyết bài toan này có thể kiểm soát bằng pool luồng, batch ML, lọc sớm, hàng đợi giới hạn và tối ưu máy chủ ảo Java JVM.

3.11.5 Kết luận tổng hợp và khuyến nghị

- Tóm tắt kết quả: MA-DIDS chính xác hơn, ít cảnh báo sai, nhưng tốn tài nguyên. Baseline IDS phản hồi nhanh, nhẹ, nhưng dễ cảnh báo sai hơn.

- Chọn MA-DIDS khi ưu tiên độ chính xác cao và phân tích sâu.

- Chọn Baseline IDS khi cần tốc độ và tiết kiệm tài nguyên.

3.12 Kết luận chương 3

Nội dung chương giới thiệu về đảm bảo an toàn thông tin, bảo mật cho công nghệ Tác tử di động, các điểm yếu của kiến trúc Tác tử di động và đề xuất các phương thức mới nâng cao an toàn bảo mật nền tảng tác tử di động qua sử dụng thuật toán M-PKI và xây dựng bộ Khung phát hiện xâm nhập hệ thống mạng dựa trên nền tảng Tác tử di động.

Nội dung của chương được tổng hợp và là kết quả của công trình đã được công bố tại [CT2], [CT4] và [CT5]. Các công trình nghiên cứu đem lại hiệu quả và đóng góp khoa học chính như sau:

- Thứ nhất, đề xuất các phương thức nâng cao an toàn bảo mật nền tảng tác tử di động qua sử dụng thuật toán mã hoá nhỏ gọn và đơn giản hơn.

- Thứ hai, đề xuất xây dựng mô hình bộ Khung phát hiện xâm nhập hệ thống mạng dựa trên nền tảng Tác tử di động.

KẾT LUẬN

Luận án này tập trung vào việc giải quyết hai vấn đề chính để xây dựng một giải pháp quản lý mạng cục bộ trong quá trình di chuyển lên mạng đám mây một cách an toàn, dựa trên công nghệ tác tử.

Vấn đề thứ nhất được giải quyết thông qua việc đề xuất một kiến trúc kết hợp hai chuẩn quản lý mạng phổ biến là SNMP và CMIP. SNMP được sử dụng để quản lý các thiết bị mạng đơn giản và thực hiện các tác vụ cơ bản, trong khi CMIP được áp dụng cho các hệ thống phức tạp yêu cầu tính bảo mật cao. Một lớp trung gian (middleware) được triển khai để tích hợp hai giao thức này, đảm bảo tính tương thích và hiệu quả trong quá trình quản lý mạng. Đồng thời, đề xuất mô hình quản lý mạng ứng dụng Công nghệ tác tử di động.

Vấn đề thứ hai liên quan đến việc tăng cường tính an toàn cho công nghệ tác tử nguồn mở. Giải pháp đề xuất bao gồm mã hóa dữ liệu bằng các thuật toán ECC thay thế cho thuật toán AES và RSA, xác thực và quản lý danh tính thông qua PKI cho nền tảng tác tử di động, và tích hợp hệ thống giám sát và phát hiện xâm nhập (IDS) ứng dụng tác tử di động để thực hiện cập nhật và vá lỗi tự động, cũng như áp dụng cơ chế quản lý quyền truy cập dựa trên vai trò (RBAC). Kết quả là một hệ thống quản lý mạng linh hoạt, an toàn, phù hợp với xu hướng chuyển đổi số và di chuyển lên Đám mây hiện nay.

Tác tử di động là một hướng nghiên cứu công nghệ mới, và được thừa nhận rộng rãi và ngay lập tức đã thu hút sự quan tâm ngày càng lớn của giới nghiên cứu cũng như giới công nghiệp trong lĩnh vực Công Nghệ Thông Tin. Trong thập kỉ đầu của thế kỷ 21, đây là khoảng thời gian bùng nổ trong lĩnh vực nghiên cứu về công nghệ tác tử nói chung và tác tử di động nói riêng, rất nhiều bộ khung tác tử được phát triển, các chuẩn về tác tử cũng được xây dựng và phát hành. Lĩnh vực nghiên cứu công nghệ tác tử rất rộng từ những lĩnh vực có thể áp dụng ngay trong thực tiễn như điện toán di động, tài liệu động, lấy dữ liệu từ xa, quản trị hệ thống mạng…cho đến những lĩnh vực mới như điện toán đám mây, điện toán môi trường bao quanh.

Mặc dù, trong lĩnh vực nghiên cứu công nghệ tác tử thu được thành tựu lớn và quan trọng nhưng việc áp dụng công nghệ vào trong thực tiễn vẫn còn ít, chưa có

nhiều phần mềm ứng dụng công nghệ tác tử nổi bật bởi các ứng dụng truyền thống (theo mô hình máy trạm/máy chủ). Nguyên nhân của tình trạng trên là sự lo ngại về công nghệ mới, tính an toàn và bảo mật của công nghệ tác tử cũng cần phải được xem xét và phát triển hơn nữa.

Do vậy, việc nghiên cứu và làm chủ công nghệ tác tử là một vấn đề rất quan trọng nói chung và trong việc xây dựng hệ thống quản lý mạng và hứa hẹn sẽ là bước đột phá trong việc xây dựng mô hình quản lý mạng cho hệ thống mạng và điện toán đám mây hiện nay. Mặt khác, ngoài việc làm chủ công nghệ tác tử thì môi trường tác tử, khung ứng dụng để triển khai tác tử và sự đảm bảo tính an toàn bảo mật cho hệ thống tác tử là một vấn đề trọng yếu để đưa công nghệ tác tử vào triển khai trong thực tế.

Kết quả chính của luận án:

Thứ nhất, nghiên cứu và đưa ra bộ điều hợp cho phép kết hợp cả 2 giao thức SNMP và CMIP cũng như đưa bộ kết nối này vào 2 giao thức này vào ứng dụng trong nền tảng tác tử di động và đề xuất mô hình quản lý mạng cho mạng cục bộ và đám mây sử dụng nền tảng tác tử di động là Mô hình CNMMA giúp quản lý hiệu quả lưu lượng đám mây với tiết kiệm chi phí, băng thông mạng hơn và cung cấp giải pháp bảo mật cho việc quản lý mạng.

Thứ hai, đề xuất các phương thức nâng cao an toàn bảo mật nền tảng tác tử di động qua sử dụng thuật toán mã hoá nhỏ gọn và đơn giản hơn và đề xuất xây dựng mô hình bộ Khung phát hiện xâm nhập hệ thống mạng dựa trên nền tảng Tác tử di động.

Hướng phát triển của luận án:

Thứ nhất, tiếp tục mở rộng, nghiên cứu nâng cao an toàn bảo mật và cải tiến kỹ thuật cho nền tảng tác tử di động để có thể ứng dụng các tác tử di động vào trong thực tế .

Thứ hai, nghiên cứu và ứng dụng Trí tuệ nhân tạo vào nền tảng tác tử di động như cải tiến về thuật toán tối ưu, và dò tìm vị trí của tác tử di động, quản lý mạng cục bộ và đám mây tự động.

DANH MỤC CÔNG TRÌNH TÁC GIẢ ĐÃ CÔNG BỐ

[CT1]. Nguyen Minh Phuc, Nguyen Ai Viet, Tran Quy Nam, Cloud network management model based on Mobile Agent, Proceeding of Conference FAIR 2020, DOI: 10.15625/vap.2020.00150, (2020).

[CT2]. Nguyen Minh Phuc, Nguyen Ai Viet, Tran Quy Nam, Enhanced security and performance of the Smart Traffic Management System VNSMAPS by using Mobile Agent and Map Reduce, Proceeding of Conference FAIR 2021, DOI: 10.15625/vap.2021.0100, (2021).

[CT3]. Nguyen Minh Phuc; Nguyen Ai Viet; Tran Quy Nam; Long Cu Kim; Vijender Kumar Solanki, "Enhanced SDN Security Using Mobile Agent" in Creative Approaches Towards Development of Computing and Multidisciplinary IT Solutions for Society, Wiley, 2024, pp.25-36, doi: 10.1002/9781394272303.ch3

[CT4]. Nguyen Minh Phuc, Nguyen Ai Viet, Tran Quy Nam, Integration of SNMP and CMIP protocol for Mobile Agent in LAN and Cloud Network Management, (2024), The International Scientific Journal Current Research # 45 (227), November 2024, International Scientific Journal Actual Research (apni.ru), ISSN 2713-1513.

[CT5]. Nguyen Minh Phuc, Nguyen Ai Viet, Tran Quy Nam, CNNMA model for enhancing security and network performance in LAN and Cloud Network Management, (2025), The International Scientific Journal Current Research #1 (287), December 2025, International Scientific Journal Actual Research (apni.ru), ISSN 2713-1513.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. Nguyễn Minh Phúc (2017), Tài liệu chuyên đề "*Nghiên cứu phát triển các giải pháp nâng cao tính an toàn và giao thức trao đổi dữ liệu quản lý mạng an toàn trên hạ tầng tác tử di động*" thuộc Dự án cấp nhà nước thuộc Chương trình Công nghệ cao Quốc gia "*Giải pháp phần mềm Quản lý nội dung thông tin từ xa dựa trên Công nghệ V-AZUR*".

[2]. Nguyễn Minh Phúc (2017), Tài liệu chuyên đề "*Nghiên cứu, đề xuất phương án làm chủ công nghệ tác tử và bộ mã nguồn mở KVM*" thuộc Dự án "*Giải pháp phần mềm Quản lý nội dung thông tin từ xa dựa trên Công nghệ V-AZUR*".

[3]. Nguyễn Minh Phúc (2017), Tài liệu chuyên đề "*Thiết kế module cấu hình, đăng ký, kiểm tra tác tử di động*" thuộc Dự án "*Giải pháp phần mềm Quản lý nội dung thông tin từ xa dựa trên Công nghệ V-AZUR*".

[4]. Nguyễn Minh Phúc (2017), Tài liệu chuyên đề "*Xây dựng đặc tả tác tử quản lý mạng LAN và mạng Cloud*" thuộc Dự án "*Giải pháp phần mềm Quản lý nội dung thông tin từ xa dựa trên Công nghệ V-AZUR*".

[5]. Đỗ Xuân Hoàng, Nguyễn Hoàng Nam (2021), *Tài liệu thiết kế hệ thống và đặc tả hệ thống phần mềm hệ thống giao thông thông minh VNSMAPS*.

[6]. PGS.TS Nguyễn Ái Việt (2014), *Bằng sáng chế công nghệ V-AZUR*.

[7]. PGS.TS Nguyễn Ái Việt (2017), Dự án "*Giải Pháp Bảo vệ an toàn mạng LAN cho cơ quan doanh nghiệp dựa trên Công Nghệ Tác Tử*".

[8]. PGS. TS Đỗ Trung Tuấn (2002), *Quản trị mạng máy tính*, Nhà xuất bản Đại học Quốc gia Hà Nội.

Tiếng Anh

[9]. A.K. Sharma1, Atul Mishra, Vijay Singh (2012), *An intelligent mobile-agent based scalable network management architecture for large-scale enterprise system*, International Journal of Computer Networks & Communications (IJCNC) Vol.4, No.1, January 2012, DOI : 10.5121/ijcnc.2012.4107 79.

[10]. Aayush Pradhana, Rejo Mathewb (2020), *Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN)*, Third International Conference on Computing and Network Communications (CoCoNet'19), Procedia Computer Science 171 2581-2589.

[11]. Abubakr S. Eltayeb, Halla O. Almubarak, Tahani Abdalla Attia (2013), *A GPS Based Traffic Light Pre-emption Control System for Emergency Vehicles*, International Conference On Computing, Electrical And Electronic.

[12]. Adam Dou, Vana Kalogeraki, Dimitrios Gunopulos, Taneli Mielikainen and Ville H. Tuulos, (2010), *Misco: A MapReduce Framework for Mobile Systems*.

[13]. Akram Hakiria, b, Aniruddha Gokhalec, Pascal Berthoua, Douglas C. Schmidtc, Gayraud Thierrya (2014), *Software-defined Networking: Challenges and Research Opportunities for Future Internet.*

[14]. Alhilali, A.H (2023), *Design and implement a real-time network traffic management system using SNMP protocol*, Eastern-European Journal of Enterprise Technologies.

[15]. Alhilali, Ahmed & Al Farawn, Ali & Mjhool, Ahmed (2023), *Design and implement a real-time network traffic management system using SNMP protocol.* Eastern-European Journal of Enterprise Technologies, 5, 35-44, 10.15587/1729-4061.2023.286528.

[16]. Alkasassbeh, Mouhammd & Adda, Mo, (2008), Analysis of mobile agents in network fault management, J. Network and Computer Applications. 31. 699-711. 10.1016/j.jnca.2007.11.005.

[17]. Almasan, P., Rusek, K., Xiao, S., Shi, X., Cheng, X., Cabellos-Aparicio, A., & Barlet-Ros, P (2023), Leveraging Spatial and Temporal Correlations for Network Traffic Compression, ArXiv, abs/2301.08962.

[18]. Almseidin, M., Alkasassbeh, M., & Kovacs, S (2018), Fuzzy Rule Interpolation and SNMP-MIB for Emerging Network Abnormality. ArXiv, abs/1811.08954, DOI:10.18517/IJASEIT.9.3.7360.

[19]. Al-Naymat, G., Al-kasassbeh, M., Al-Hawari, E. (2019), *Exploiting SNMP-MIB Data to Detect Network Anomalies Using Machine Learning Techniques. In: Arai, K., Kapoor, S., Bhatia, R. (eds) Intelligent Systems and Applications,* IntelliSys 2018, Advances in Intelligent Systems and Computing, vol 869. Springer, Cham. DOI: doi.org/10.1007/978-3-030-01057-77 3.

[20]. Amazon Web Services (2021), Shared Responsibility Model.

[21]. Amir Vahid Dastjerdi, Kamalrulnizam Abu Bakar, Sayed Gholam Hassan Tabatabaei (2009), Distributed Intrusion Detection in Clouds Using Mobile Agents, Third International Conference on Advanced Engineering Computing and Applications in Sciences.

[22]. Anish Saini, & Atul Mishra (2014), Domain-partitioned element management Systems employing mobile agents for Distributed network management, International Journal of Computer Networks & Communications (IJCNC) Vol.6, No.3, May 2014, DOI : 10.5121/ijcnc.2014.6309 107.

[23]. Barreiros, M. and Lundqvist, P (2016), QOS-Enabled Networks: Tools and Foundations, Second Edition, Print ISBN:9781119109105. Online ISBN: 9781119109136, DOI:10.1002/9781119109136.

[24]. Belal Amro (2014), *Mobile Agent Systems*, Recent Security Threats and Counter Measures.

[25]. Breaban, M. C. et al (2018), *Bandwidth management application in directory service environment'*, 2018 14th International Conference on Development and Application Systems, DAS 2018-Proceedings, pp. 88–92. doi: 10.1109/DAAS.2018 .8396077.

[26]. CCITT (1991*), X.710 and ISO/IEC 9595: Information technology - Open Systems Interconnection - Common management information service definition*.

[27]. Chen, L., Li, Q. (2023), *Performance Benchmarking of Agent-Based vs. Centralized Cloud Monitoring*, Journal of Network and Systems Management, vol. 31, no. 3, pp. 1-25, DOI: 10.1007/s10922-023-09750-3.

[28]. Cloud Security Alliance (2019), *Security Guidance for Critical Areas of Focus in Cloud Computing v4.0*.

[29]. Dabbagh, M., Hamdaoui, B., Guizani, M., & Rayes (2015), *Software-Defined Networking Security: Pros and Cons*; IEEE Communications Magazine, 53(6), 73-79, doi:10.1109/MCOM.2015.7120048.

[30]. Diana, F. M (2016), *Implementasi Simple Network Manegement Protocol (SNMP)*, pada aplikasi monitoring Jaringan.

[31]. Dierks, T., & Rescorla, E (2008), *The Transport Layer Security (TLS) Protocol Version 1.2*, IETF RFC 5246.

[32]. Dinh, H. T., Lee, C., & Niyato, D (2013), *A survey of mobile cloud computing: Architecture, applications and approaches*, Wireless Communications and Mobile Computing, 13(18), 1587-1611.

[33]. Divakara K. Udupa, McGraw-Hill (2004), *Telecommunications Management Network.*

[34]. Dr. Mamta Madan and Mohit Mathur (2014), *Cloud network management model - A novel approach to manage cloud traffic*, International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol. 4, No. 5, October 2014, DOI : 10.5121/ijccsa.2014.4502 9.

[35]. ENISA (2018), *Cloud Computing Risk Assessment*.

[36]. Espinel Villalobos, R.I., Ardila Triana, E., Zarate Ceballos, H., & Ortiz Triviño, J.E (2021), *Design and Implementation of Network Monitoring System for Campus Infrastructure Using Software Agents*, Ingeniería e Investigación.

[37]. Fabio Bellifemine, Giovanni Caire, Dominic Greenwood (2007), *Developing Multi-Agent Systems with JADE*, John Wiley & Sons Ltd.

[38]. Ferguson, N., & Schneier, B (2003), *Practical Cryptography*, Wiley.

[39]. Fernandes, D. A. B., Soares, L. F. B., & Gomes, J. V (2014), *Security issues in cloud environments: A survey*, International Journal of Information Security, 13(2), 113-170.

[40]. Ferrag, M.A. et al. (2022), *SDN-NFV Orchestration for Secure Multi-Cloud Migration*, IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3124-3139, DOI: 10.1109/TDSC.2022.3185172.

[41]. FIPA (2002), *FIPA ACL Message Structure Specification - SC00061G*.

[42]. Forrester Research (2019), *Zero Trust Extended Ecosystem Platform Vendors*, Q4 2019.

[43]. García, E. et al. (2022), *FIPA-Compliant Mobile Agent Platforms: Survey and Performance Analysis*, ACM Computing Surveys, vol. 55, no. 8, pp. 1-37,DOI: 10.1145/3579353.

[44]. Gavalas, Damianos & Greenwood, Dominic & Ghanbari, Mohammed (2002), *Hierarchical Network Management: A Scalable and Dynamic Mobile Agent in Network Fault Management*.

[45]. Gholamreza Farahani, Rahani, Gholamreza (2017), *New proposed architecture for Q3 interface to manage IP-based networks*, International Journal of Computer Networks & Communications (IJCNC), Vol.9, No.4, July 2017.

[46]. Hashizume, K., Rosado, D. G., & Fernández-Medina, E (2013), *An analysis of security issues for cloud computing*, Journal of Internet Services and Applications, 4(1), 1-13.

[47]. Howard, M., & LeBlanc, D (2003), *Writing Secure Code*, Microsoft Press.

[48]. I. Ahmad, S. Namal, M. Ylianttila and A. Gurtov (2015), *Security in Software Defined Networks: A Survey*, IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2317-2346, Fourthquarter.

[49]. Ichiro Satoh (2001), *MobiDoc: A Mobile Agent-based Framework for Compound Documents*, Informatica, vol.25, no. 4, pp.493-500.

[50]. Ichiro Satoh (2003), *A Testing Framework for Mobile Computing Software*, IEEE Transactions on Software Engineering, vol. 29, no. 12, pp.1112-1121.

[51]. Ichiro Satoh (2003), *Building Reusable Mobile Agents for Network Management*, IEEE Transactions on Systems, Man and Cybernetics, vol.33, no. 3, part-C, pp.350-357.

[52]. Ichiro Satoh (2003), *SpatialAgents: Integrating User Mobility and Program Mobility in Ubiquitous Computing Environments*, Wireless Communications and Mobile Computing, vol.3, no.4, pp.411-423, John Wiley.

[53]. Ichiro Satoh (2004), *Selection of Mobile Agents*, Proceedings of 24th IEEE International Conference on Distributed Computing Systems (ICDCSb2004), pp.484-493, IEEE Computer Society.

[54]. Ichiro Satoh (2004*), Software Testing for Wireless Mobile Computing*, IEEE Wireless Communications, vol. 11,no. 5, pp.58-64, IEEE Communication Society.

[55]. Ichiro Satoh (2006), *Building and Selecting Mobile Agents for Network Management*, Journal of Network and Systems Management, vol.14, no.1, pp.147-169, Springer.

[56]. Ichiro Satoh (2006), *Mobile Agents*, In: Scerri, P., Vincent, R., Mailler, R. (eds) Coordination of Large-Scale Multiagent Systems. Springer, Boston, MA. https://doi.org/10.1007/0-387-27972-5_11.

[57]. IEEE-SA Standards Board, LAN/MAN Standards Committee of the IEEE Computer Society (2011), *IEEE Standard for Management Information Base (MIB) Definitions for Ethernet*.

[58]. Ismail, Leila & Hagimont, Daniel (1999), *A Performance Evaluation of the Mobile Agent Paradigm*, Sigplan Notices - SIGPLAN. 34. 306-313. 10.1145/320384.320415.

[59]. ISO/IEC (2013), *ISO/IEC 27001:2013 - Information Security Management*.

[60]. ISO/IEC (2015), *ISO/IEC 27002:2015 - Code of practice for information security controls*.

[61]. J. Swarna, C. Senthil raja, Dr.K.S.ravichandran (2012), *Cloud monitoring based on SNMP*.

[62]. Jeffrey Dean and Sanjay Ghemawat (2004), *MapReduce: Simplified Data Processing on Large Clusters*.

[63]. Koerner, E. (1997), *Design of a proxy for managing CMIP agents via SNMP*, Comput. Commun., 20, 349-360.

[64]. Kotz, D., Gray, R. S., & Rus, D (2002), *Future directions for mobile agent research*, IEEE Distributed Systems Online, 3(8).

[65]. Kumar, R. et al. (2023), *MAFIA: Mobile Agent Framework for Intent-Based Networking*, Computer Networks, vol. 225, pp. 109851, DOI: 10.1016/j.comnet.2023.10985.

[66]. Lange, D. B., & Oshima, M (1999), *Seven good reasons for mobile agents*, *Communications of the ACM*, 42(3), 88-89.

[67]. Laurent Andrey, Olivier Festor, Abdelkader Lahmadi, Aiko pras and Jürgen Schönwälder (2009), *Survey of SNMP performance analysis studies*, International Journal of Network Management, 19 pp. 527-54.

[68]. Le, M., Doan Huu, H., Nguyen Ngoc, T., Cu Kim, L., & Nguyen Minh, P (2017*), An Assessment Model for Cyber Security of Vietnamese Organization*, VNU Journal Of Science: Policy And Management Studies, 33(2). doi:10.25073/2588-1116/vnupam.4102.

[69]. Madhukrishna Priyadarsini, Padmalochan Bera (2021), *Software defined networking architecture, traffic management, security, and placement: A survey*, Computer Networks,Volume 192, 108047,ISSN 1389-1286, doi: doi.org/10.1016/j.comnet.2021.108047.

[70]. Mell, P., & Grance, T (2011), *The NIST definition of cloud computing*, NIST Special Publication 800-145.

[71]. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A (1996), *Handbook of Applied Cryptography*, CRC Press.

[72]. Mohit Dev Srivastava, Prerna, Shubhendu Sachin, Sumedha Sharma, Utkarsh Tyagi (2012), *Smart Traffic Control System using PLC and SCADA*, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 1, Issue 2, December 2012.

[73]. Montaño, M., Torres, R., Ludeña, P., Sandoval, F (2021), *IoT Management Analysis Using SDN: Survey*, In: Botto-Tobar, M., Montes León, S., Camacho, O., Chávez, D., Torres-Carrión, P., Zambrano Vizuete, M. (eds) Applied Technologies. Communications in Computer and Information Science, vol 1388. Springer, Cham. doi: doi.org/10.1007/978-3-030-71503-8_45.

[74]. Muller, N.J (2007), *Integrated network management, Information Systems Management*, DOI:10.1080/10580539208906893.

[75]. Nguyen, T., Le, P. (2022), *Blockchain-Secured Mobile Agents for IoT Device Management*, ACM/IEEE Symposium on Edge Computing (SEC), pp. 112-128, DOI: 10.1145/3543507.3583362.

[76]. NIST (2012), *SP 800-61 Revision 2: Computer Security Incident Handling Guide*.

[77]. Paul Almasan, Krzysztof Rusek, Shihan Xiao, Xiang Shi, Xiangle Cheng, Albert Cabellos-Aparicio, Pere Barlet-Ros (2023), *Leveraging Spatial and Temporal Correlations for Network Traffic Compression*.

[78]. PCI Security Standards Council (2021), *PCI DSS Quick Reference Guide*.

[79]. Peltier, T. R (2016), *Information Security Policies, Procedures, and Standards: Guidelines for Effective Information Security Management*, CRC Press.

[80]. Pfleeger, C. P., & Pfleeger, S. L (2012), *Analyzing Computer Security: A Threat/Vulnerability/Countermeasure Approach*, Prentice Hall.

[81]. Pham, V. A., & Karmouch, A (1998), *Mobile software agents: An overview*, IEEE Communications Magazine, 36(7), 26-37.

[82]. Priyanka Sharma, Dr. Anjana Goen (2018), *Smart Traffic Control System Using Weighted Data*, International Journal of Advance Research in Science and Engineering, Volume No.07, Issue No.06, June 2018.

[83]. Sandhu, R. S., & Samarati, P (1994), *Access Control: Principles and Practice*, IEEE Communications Magazine.

[84]. Sangram Ray, G. P. Biswas (2013), *Design of Mobile Public Key Infrastructure (M-PKI) using Elliptic Curve Cryptography*, International Journal on Cryptography and Information Security, Vol.3, No.1, March 2013.

[85]. Saydam, T., & Sirsikar, R.(1998), *Design of CMIP-SNMPv2 Proxy Gateway for Network Management Interoperability*, Journal of Network and Systems Management, 6, 157-178.

[86]. Scarfone, K., & Mell, P (2007), *Guide to Intrusion Detection and Prevention Systems (IDPS)*, NIST Special Publication 800-94.

[87]. Shao-Chun Zhong, Qingfeng Song, Xiao-Chun Cheng, Yan Wang (2003), *A safe mobile agent system for distributed intrusion detection*, Proceedings of 'the Second International Conference on Machine Learning and Cybernetics, Wan, 2-5 November 2003,

[88]. Sharma, Ashok & Mishra, Atul & Singh, Vijay (2012), *An Intelligent Mobile-Agent Based Scalable Network Management Architecture for Large-Scale Enterprise System*, International Journal of Computer Networks & Communications. 4. 10.5121/ijcnc.2012.4107.

[89]. Simsek, G., Ergenç, D., & Onur, E (2022), *Reliable and Distributed Network Monitoring via In-band Network Telemetry*, ArXiv, abs/2212.14876.

[90]. SO/IEC (1997), *Common Management Information Protocol (CMIP)*, ISO/IEC 9596-1.

[91]. Soway Tech Limited, (2018*), F3- Vehicle terminal & peripheral sensors protocol v3.10*.

[92]. Stallings, W (2007), *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison-Wesley.

[93]. Stallings, W (2017), *Network Security Essentials: Applications and Standards*, Pearson.

[94]. T. D. Nadeau, K. Gray (2013), *SDN: software defined networks*, O'Reilly Media, Inc.

[95]. Tian, YC., Gao, J (2024), *Network Management Architecture. In: Network Analysis and Architecture*, Signals and Communication Technology, Springer, Singapore, Doi: 10.1007/978-981-99-5648-7_9.

[96]. Tian-jun, Z (2007), Research on nesting mobile Agent network management, Computer Engineering and Design.

[97]. Tuli, Ruchi (2023), *Analyzing Network performance parameters using wireshark*, 10.48550/arXiv.2302.03267.

[98]. Uyless Black, McGraw-Hill (1995), Second Edition, *Network Management Standard*.

[99]. Vikas Tyagi, Samayveer Singh (2023), *Network resource management mechanisms in SDN enabled WSNs: A comprehensive review*, Computer Science Review, Volume 49, (2023),100569,ISSN 1574-0137, doi://doi.org/10.1016/j.cosrev.2023.100569.

[100]. Wang, Y. et al. (2023), *Unified Management of SNMP-CMIP Networks via Graph Neural Networks*, IEEE INFOCOM, pp. 1-10, DOI: 10.1109/INFOCOM53939.2023.10228937.

[101]. Wenfeng Xia, Yonggang Wen, Dusit, Niyato (2015), *A Survey on Software-Defined*, Networking IEEE Communications Survery& Tutorials, vol 17, No.1, First Quarter 2015, A Survey on Software-Defined Networking.

[102]. Whitman, M. E., & Mattord, H. J (2018), *Principles of Information Security, Cengage Learning*.

[103]. Ya-shiang peng, Yen-cheng Chen (2011), *SNMP-based monitoring of Heterogeneous virtual infrastructure in Clouds*.

[104]. Zaid Ibrahim Rasool1, Ridhab Sami Abd Ali1 and Musaddak Maher Abdulzahra (2020), *Network Management in Software-Defined Network: A Survey*, *Materials Science and Engineering*, Volume 1094, 1st International Conference on Sustainable Engineering and Technology (INTCSET 2020) 15th-16th December 2020, Baghdad, Iraq, DOI 10.1088/1757-899X/1094/1/012055.

[105]. Zhang, K., Liu, M. (2023), *Adaptive Mobile Agents for Distributed Network Monitoring in 5G/6G Networks*, IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1450-1465,DOI: 10.1109/TNSM.2023.3012287.

[106]. Zihang, R., & Lobelle, M (1994), *Network management integrating SNMP/CMIP protocol implementations*, Annales Des Télécommunications, 49, 17-26. DOI:10.1007/BF02999639.

[107]. Zubair, M., & Manzoor, U (2016), *Mobile agent based network management applications and fault-tolerance mechanisms*, 2016 Sixth International Conference on Innovative Computing Technology (INTECH), 441-446.

PHỤ LỤC

Code thực nghiệm xây dựng bảng ánh xạ đối tượng trong giao thức CMIP và SNMP và các nền tảng tác tử bằng ngôn ngữ Java

1. GatewayApplication.java – executeCMIPToSNMPMapping

```
// GatewayApplication.java
public CompletableFuture<Object> executeCMIPToSNMPMapping(
    String operation, String connection,
    String objectClass, String objectInstance,
    Map<String, Object> attributes) {
    Map<String, Object> request = new HashMap<>();
    request.put("operation", operation);
    request.put("targetName", connection);
    request.put("objectClass", objectClass);
    request.put("objectInstance", objectInstance);
    request.put("attributes", attributes);
    return mappingEngine.executeMapping(
        "CMIP_" + operation.toUpperCase() + "_TO_SNMP_GET",
        request
    );
}
```

2. ProtocolMappingEngine.java – findObjectTypeMappingByCMIPClass

```
// ProtocolMappingEngine.java
private ObjectTypeMapping findObjectTypeMappingByCMIPClass(String
objectClass) {
    for (ObjectTypeMapping mapping : objectTypeMappings.values())
{
        if (mapping.getCmipObjectClass().equals(objectClass)) {
            return mapping;
        }
    }
    return null;
}
```

3. ProtocolMappingEngine.java – mapCMIPAttributesToSNMP

```
// ProtocolMappingEngine.java
private List<String> mapCMIPAttributesToSNMP(
    List<String> cmipAttributes,
    ObjectTypeMapping mapping) {
    List<String> snmpOids = new ArrayList<>();
    for (String cmipAttribute : cmipAttributes) {
        for (AttributeMapping attrMapping :
mapping.getAttributeMappings().values()) {
```

```
            if
(attrMapping.getCmipAttribute().equals(cmipAttribute)) {
                snmpOids.add(attrMapping.getSnmpOid());
            }
        }
    }
    return snmpOids;
}
```

4. ProtocolMappingEngine.java – mapCMIPGetToSNMPGet

```
// ProtocolMappingEngine.java - mapCMIPGetToSNMPGet()
private CompletableFuture<SNMPResponse> mapCMIPGetToSNMPGet(
    Map<String, Object> cmipRequest) {
    return CompletableFuture.supplyAsync(() -> {
        try {
            String objectClass = (String)
cmipRequest.get("objectClass");
            String objectInstance = (String)
cmipRequest.get("objectInstance");
            List<String> attributes = (List<String>)
cmipRequest.get("attributes");
            String targetName = (String)
cmipRequest.get("targetName");
            ObjectTypeMapping mapping =
findObjectTypeMappingByCMIPClass(objectClass);
            if (mapping == null) {
                return new SNMPResponse(false, "No mapping found
for CMIP object class: " + objectClass, null);
            }
            List<String> snmpOids =
mapCMIPAttributesToSNMP(attributes, mapping);
            Map<String, Object> results = new HashMap<>();
            for (String oid : snmpOids) {
                SNMPResponse snmpResponse =
snmpGateway.get(targetName, oid).join();
                if (snmpResponse.isSuccess()) {
                    results.put(oid, snmpResponse.getData());
                }
            }
            return new SNMPResponse(true, "CMIP M-GET mapped to
SNMP GET", results);
        } catch (Exception e) {
            logger.error("Error mapping CMIP M-GET to SNMP GET",
e);
            return new SNMPResponse(false, e.getMessage(), null);
        }
```

```
    });
}
```

5. ProtocolMappingEngine.java – findObjectTypeMappingByOID

```
// ProtocolMappingEngine.java
private ObjectTypeMapping findObjectTypeMappingByOID(String oid) {
    for (ObjectTypeMapping mapping : objectTypeMappings.values())
{
        if (oid.startsWith(mapping.getSnmpOid())) {
            return mapping;
        }
    }
    return null;
}
```

6. ProtocolMappingEngine.java – extractObjectInstance

```
// ProtocolMappingEngine.java
private String extractObjectInstance(String oid, String baseOid) {
    if (oid.length() > baseOid.length()) {
        return oid.substring(baseOid.length() + 1);
    }
    return "0";
}
```

7. ProtocolMappingEngine.java – mapSNMPAttributesToCMIP

```
// ProtocolMappingEngine.java
private List<String> mapSNMPAttributesToCMIP(
    String oid, ObjectTypeMapping mapping) {
    List<String> cmipAttributes = new ArrayList<>();
    for (AttributeMapping attrMapping :
mapping.getAttributeMappings().values()) {
        if (oid.startsWith(attrMapping.getSnmpOid())) {
            cmipAttributes.add(attrMapping.getCmipAttribute());
        }
    }
    return cmipAttributes;
}
```

8. ProtocolMappingEngine.java – mapSNMPGetToCMIPGet

```
// ProtocolMappingEngine.java - mapSNMPGetToCMIPGet()
private CompletableFuture<CMIPResponse> mapSNMPGetToCMIPGet(
    Map<String, Object> snmpRequest) {
    return CompletableFuture.supplyAsync(() -> {
        try {
            String oid = (String) snmpRequest.get("oid");
```

```
            String targetName = (String)
snmpRequest.get("targetName");
            ObjectTypeMapping mapping =
findObjectTypeMappingByOID(oid);
            if (mapping == null) {
                return new CMIPResponse(false, "No mapping found
for OID: " + oid, null);
            }
            String objectInstance = extractObjectInstance(oid,
mapping.getSnmpOid());
            List<String> cmipAttributes =
mapSNMPAttributesToCMIP(oid, mapping);
            return cmipGateway.mGet(
                targetName,
                mapping.getCmipObjectClass(),
                objectInstance,
                cmipAttributes
            ).join();
        } catch (Exception e) {
            logger.error("Error mapping SNMP GET to CMIP M-GET",
e);
            return new CMIPResponse(false, e.getMessage(), null);
        }
    });
}
```

9. Mapping configuration classes

```
// Mapping configuration
public class MappingRule {
    private String ruleName;
    private String sourceProtocol;
    private String targetProtocol;
    private String sourceOperation;
    private String targetOperation;
    private Map<String, Object> conditions;
    private Map<String, Object> transformations;
}
public class ObjectTypeMapping {
    private String snmpOid;
    private String cmipObjectClass;
    private Map<String, AttributeMapping> attributeMappings;
}
public class AttributeMapping {
    private String snmpOid;
    private String cmipAttribute;
    private String dataType;
```

```
    private Map<String, Object> conversionRules;
}
```

10. ProtocolMappingEngine.java – convertValue

```
// Value conversion
private Object convertValue(String value, Integer snmpSyntax,
String cmipDataType) {
    try {
        switch (cmipDataType.toLowerCase()) {
            case "integer": return Integer.parseInt(value);
            case "string": return value;
            case "timeticks": return Long.parseLong(value);
            case "counter": return Long.parseLong(value);
            case "boolean": return Boolean.parseBoolean(value);
            default: return value;
        }
    } catch (Exception e) {
        logger.warn("Error converting value: {}", value, e);
        return value;
    }
}
```

11. GatewayMobileAgent.java – setup & capabilities

```
// GatewayMobileAgent.java
public class GatewayMobileAgent extends Agent {
    @Override
    protected void setup() {
        logger.info("Setting up Gateway Mobile Agent");
        initializeAgentData();
        addBehaviour(new MessageHandlingBehaviour());
        addBehaviour(new ProtocolMappingBehaviour());
        addBehaviour(new MonitoringBehaviour());
        registerCapabilities();
    }
    private void initializeAgentData() {
        agentData.put("agentId", getAID().getName());
        agentData.put("startTime", System.currentTimeMillis());
        agentData.put("state", AgentState.ACTIVE);
        agentData.put("capabilities", getAgentCapabilities());
        agentData.put("protocolMappings", new
ConcurrentHashMap<>());
    }
    private Map<String, Object> getAgentCapabilities() {
        Map<String, Object> capabilities = new HashMap<>();
        capabilities.put("snmpSupport", true);
        capabilities.put("cmipSupport", true);
```

```
        capabilities.put("protocolMapping", true);
        capabilities.put("eventHandling", true);
        capabilities.put("monitoring", true);
        return capabilities;
    }
}
```

12. MessageHandlingBehaviour – receive & dispatch

```
// MessageHandlingBehaviour
private class MessageHandlingBehaviour extends CyclicBehaviour {
    @Override
    public void action() {
        MessageTemplate template =
MessageTemplate.MatchPerformative(ACLMessage.INFORM);
        ACLMessage msg = receive(template);
        if (msg != null) {
            handleMessage(msg);
        } else {
            block();
        }
    }
    private void handleMessage(ACLMessage msg) {
        String content = msg.getContent();
        String conversationId = msg.getConversationId();
        if (content.startsWith("SNMP_REQUEST:")) {
            handleSNMPRequest(content, conversationId);
        } else if (content.startsWith("CMIP_REQUEST:")) {
            handleCMIPRequest(content, conversationId);
        } else if (content.startsWith("MAPPING_REQUEST:")) {
            handleMappingRequest(content, conversationId);
        }
    }
}
```

13. ProtocolMappingBehaviour – mapping loop

```
// ProtocolMappingBehaviour
private class ProtocolMappingBehaviour extends CyclicBehaviour {
    @Override
    public void action() {
        Queue<MappingRequest> requests =
getPendingMappingRequests();
        for (MappingRequest request : requests) {
            if (request.isSNMPToCMIP()) {
                performSNMPToCMIPMapping(request);
            } else if (request.isCMIPToSNMP()) {
                performCMIPToSNMPMapping(request);
```

```
            }
        }
        block(5000);
    }
}
```

14. MobileAgentPlatform.java – platform & agent creation

```
// MobileAgentPlatform.java
public class MobileAgentPlatform {
    private Runtime runtime;
    private AgentContainer mainContainer;
    private Map<String, AgentController> agents;
    public void startPlatform(String platformName, String host,
int port) {
        Profile profile = new ProfileImpl();
        profile.setParameter(Profile.PLATFORM_ID, platformName);
        profile.setParameter(Profile.MAIN_HOST, host);
        profile.setParameter(Profile.MAIN_PORT,
String.valueOf(port));
        profile.setParameter(Profile.GUI, "false");
        runtime = Runtime.instance();
        mainContainer = runtime.createMainContainer(profile);
        logger.info("Mobile Agent Platform started: {}",
platformName);
    }
    public String createGatewayAgent(String agentName) {
        AgentController agentController =
mainContainer.createNewAgent(
            agentName,
            GatewayMobileAgent.class.getName(),
            null
        );
        agentController.start();
        agents.put(agentName, agentController);
        logger.info("Mobile agent created: {}", agentName);
        return agentName;
    }
}
```

15. JADE ACL – sending a request

```
// ACL message example
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID("protocol-mapper-agent",
AID.ISLOCALNAME));
msg.setContent("SNMP_REQUEST:GET:1.3.6.1.2.1.1.1.0");
msg.setConversationId("conv-12345");
```

```
msg.setLanguage("text");
msg.setOntology("protocol-mapping");
send(msg);
```

16. Agent migration – doMove()

```
// Agent migration
public void migrateToPlatform(String platformName) {
    addBehaviour(new OneShotBehaviour() {
        @Override
        public void action() {
            try {
                logger.info("Migrating agent to platform: {}",
platformName);
                doMove(new jade.core.Location(platformName,
null));
            } catch (Exception e) {
                logger.error("Error during agent migration", e);
            }
        }
    });
}
```