BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Huỳnh Nguyên Chính

GIẢI PHÁP PHÁT HIỆN NHANH CÁC HOT-IP TRONG HỆ THỐNG MẠNG VÀ ỨNG DỤNG

LUẬN ÁN TIẾN SĨ KỸ THUẬT

HÀ NỘI – 2017

BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Huỳnh Nguyên Chính

GIẢI PHÁP PHÁT HIỆN NHANH CÁC HOT-IP TRONG HỆ THỐNG MẠNG VÀ ỨNG DỤNG

Chuyên ngành: **Hệ thống thông tin**

Mã số: 62.48.01.04

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Người hướng dẫn khoa học:

1. **PGS.TS. NGUYỄN ĐÌNH THÚC**
2. **TS. TÂN HẠNH**

HÀ NỘI – 2017

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu do tôi thực hiện. Các số liệu và kết quả trình bày trong luận án là trung thực, chưa được công bố bởi bất kỳ tác giả nào khác. Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng trong danh mục các tài liệu tham khảo.

Tác giả luận án

Huỳnh Nguyên Chính

LỜI CẢM ƠN

Trong quá trình hoàn thành luận án này, tôi đã được quý thầy cô nơi cơ sở đào tạo giúp đỡ tận tình, cơ quan nơi công tác tạo mọi điều kiện thuận lợi, bạn bè cùng gia đình thường xuyên động viên khích lệ.

Luận án này không thể hoàn thành tốt nếu không có sự tận tình hướng dẫn và sự giúp đỡ quý báu của PGS.TS Nguyễn Đình Thúc và TS. Tân Hạnh. Tôi xin được bày tỏ lòng biết ơn sâu sắc nhất đến hai thầy.

Tôi xin chân thành cảm ơn lãnh đạo Học viện Công nghệ Bưu chính Viễn thông, Khoa Quốc tế và Đào tạo sau đại học đã tạo điều kiện thuận lợi, hỗ trợ hoàn thành các thủ tục để giúp tôi hoàn thành được luận án của mình.

Cuối cùng, tôi xin cảm ơn tất cả bạn bè và người thân đã đóng góp nhiều ý kiến thiết thực và có những lời động viên khích lệ quý báu giúp tôi hoàn thành tốt luận án.

Hà Nội, tháng 04 năm 2017

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Thuật ngữ	Diễn giải tiếng Anh	Diễn giải tiếng Việt
AGT	Adaptive Group Testing	Thử nhóm ứng biến
AS	Autonomous System	Hệ thống tự trị
BGP	Border Gateway Protocol	Giao thức định tuyến BGP
Bps	Bits per second	Số lượng bit/giây
CGT	Combinatorial Group Testing	Thử nhóm tổ hợp
CMH	Count-Min	Count-Min
CS	Count-Sketch	Count-Sketch
DDoS	Distributed Denial of Service	Từ chối dịch vụ phân tán
DoS	Denial of Service	Từ chối dịch vụ
F	Frequent	Thuật toán Frequent thuộc loại Counter-based
HTTP	Hyper Text Transfer Protocol	Giao thức truyền siêu văn bản
ICMP	Internet Control Message Protocol	Giao thức bản tin điều khiển Internet
IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
IP	Internet Protocol	Địa chỉ của các thiết bị trên mạng
IPS	Intrusion Prevention System	Hệ thống phòng chống xâm nhập

ISP	Internet Service Provider	Nhà cung cấp dịch vụ Internet
LCD	Lossy Counting	Thuật toán LossyCounting thuộc loại counter-based
MDS	Maximun Distance Separable	Phân ly khoảng cách tối đa
MIMD	Multiple Instruction Stream, Multiple data stream	Kiến trúc song song, nhiều lệnh khách nhau có thể đồng thời xử lý nhiều dữ liệu khác nhau trong cùng thời điểm
MISD	Multiple Instruction Stream, Single data stream	Kiến trúc song song, nhiều lệnh cùng thao tác trên một dữ liệu
NAGT	Non-Adaptive Group Testing	Thử nhóm bất ứng biến
OSI	Open Systems Interconnection	Mô hình tham chiếu OSI
Pps	Packets per second	Số lượng gói tin/giây
PVM	Parallel Virtual Machine	Máy ảo song song
RPC	Remote Procedure Call	Lời gọi thủ tục từ xa
RS	Reed-Solomon	Reed-Solomon
SIMD	Single Instruction stream, Multiple data stream	Kiến trúc song song, một lệnh được thực hiện đồng thời trên các dữ liệu khác nhau
SISD	Single Intruction stream, single data stream	Kiến trúc song song, tại mỗi thời điểm chỉ một lệnh được thực hiện

SNMP	Single Network Management Protocol	Giao thức quản trị mạng đơn giản
SSH	Space Saving Heap	Thuật toán SpaceSaving sử dụng cấu trúc Heap
SSL	Space Saving Link	Thuật toán SpaceSaving sử dụng danh sách liên kết
URL	Uniform Resource Locator	Thuật ngữ dùng để chỉ tên của trang Web cần truy cập

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC KÝ HIỆU

Ký hiệu	Ý nghĩa
C_{in}	Mã trong
C_{out}	Mã ngoài
$C_{out} \circ C_{in}$	Phép nối mã
$c_{t\times 1}$	Vector bộ đếm
d	Số lượng Hot-IP tối đa
dist(C)	Khoảng cách mã
$\mathbb{F}_q$	Trường hữu hạn của q phần tử
f_i	Tần suất xuất hiện của IP_i trong khoảng Δ
I_q	Mã đơn vị
$[l]$	Tập các phần tử $\{1,2,...,l\}$
m	Số lượng gói tin trong một chu kỳ thuật toán
m_{ij}	Phần tử của ma trận ở hàng i và cột j của ma trận, $m_{ij} \in \{0,1\}$
$M_{t\times N}$	Ma trận nhị phân kích thước t hàng và N cột
N	Số lượng địa chỉ IP phân biệt, số cột của ma trận d-phân-cách
$[n,k]_q$	Mã tuyến tính với độ dài n, số chiều k trên trường $\mathbb{F}_q$
q	Lũy thừa của một số nguyên tố
$r_{t\times 1}$	Vector kết quả của phép thử, $r_i \in \{0,1\}^t$.
t	Số hàng của ma trận d-phân-cách, số lượng nhóm thử
δ	Ngưỡng tần suất cao

Δ	Khoảng thời gian trong một chu kỳ thuật toán
ϕ	Tham số trong chọn ngưỡng ($0 \le \phi \le 1$)

MỞ ĐẦU

1. GIỚI THIỆU

Hệ thống mạng máy tính và các ứng dụng trên mạng Internet phát triển ngày càng nhanh, đáp ứng và tạo ra môi trường rộng lớn ảnh hưởng đến nhiều lĩnh vực trong cuộc sống. Nâng cao hiệu quả hoạt động của hệ thống mạng để cung cấp dịch vụ ngày càng phong phú, đa dạng, nhanh chóng với chất lượng dịch vụ tốt hơn và an toàn là những vấn đề được đặt ra cho các nhà cung cấp dịch vụ, các nhà quản trị hệ thống mạng.

Xuất phát từ thực tế như vậy, các giải pháp phát hiện sớm các đối tượng có khả năng gây nguy hại trên mạng, nhất là hệ thống mạng trung gian ở phía các nhà cung cấp dịch vụ, có ý nghĩa quan trọng trong việc giúp giảm thiểu các ảnh hưởng xấu cho các máy chủ của khách hàng và các dịch vụ trên mạng Internet. Phát hiện sớm các đối tượng này để tiến hành các giải pháp ứng phó, ngăn chặn kịp thời là vấn đề quan trọng trong bài toán an ninh mạng.

Các gói tin lưu thông trên mạng IP đều có gắn thông tin về địa chỉ IP trong phần IP-header để xác định máy gửi và nhận. Dựa trên thông tin các địa chỉ IP, bài toán phát hiện các đối tượng hoạt động với tần suất cao trong một khoảng thời gian ngắn được gọi là bài toán phát hiện các Hot-IP.

Luận án nghiên cứu và đề xuất giải pháp phát hiện các Hot-IP trên mạng, đặc biệt là các mạng có số lượng người dùng và tần suất sử dụng rất lớn, nhằm mục đích phát hiện sớm các đối tượng có khả năng gây hại. Các Hot-IP có thể là các mục tiêu hay nguồn phát trong các tấn công từ chối dịch vụ, có thể là các máy đang tiến hành quét mạng để tìm kiếm lỗ hổng và phát tán sâu Internet, có thể là các thiết bị hoạt động bất thường trong hệ thống mạng. Phát hiện sớm các Hot-IP là bước cơ bản và quan trọng đầu tiên, từ đó giúp người quản trị xác định và tiến hành các giải pháp phòng chống hiệu quả, kịp thời.

2. LÝ DO CHỌN ĐỀ TÀI

Lưu lượng mạng và tốc độ truy cập mạng ngày một tăng cao. Điều này, một mặt mang lại rất nhiều lợi ích cho người sử dụng, mặt khác lại là nguy cơ của các tấn công mạng. Một trong các dạng tấn công rất nguy hiểm là tấn công từ chối dịch vụ (DoS), đặc biệt là tấn công từ chối dịch vụ phân tán (DDoS) (gọi chung là tấn công từ chối dịch vụ), các máy quét mạng tìm kiếm lỗ hổng để phát tán sâu Internet. Đặc trưng quan trọng của các dạng tấn công này là số lượng gói tin mang các đối tượng tấn công rất lớn trong dòng các gói tin IP xuất hiện trong khoảng thời gian rất ngắn. Phát hiện sớm các Hot-IP, những IP xuất hiện với tần suất cao trong một khoảng thời gian xác định, là bước quan trọng đầu tiên để xác định các đối tượng có khả năng gây nguy hại trên mạng. Trong các mạng với số lượng rất lớn các gói tin lưu thông như mạng trung gian của các nhà cung cấp dịch vụ cần phân tích và xử lý các dòng dữ liệu thời gian thực, các giải pháp phát hiện và phòng chống phải đơn giản, nhanh chóng và hiệu quả.

Trong các nghiên cứu liên quan đến phát hiện và phòng chống tấn công từ chối dịch vụ, căn cứ thời điểm tấn công các nhà nghiên cứu chia thành ba giai đoạn tương ứng liên quan đến việc phòng chống: *đề phòng* (trước khi tấn công), *phát hiện và phòng chống* (trong quá trình tấn công), *truy tìm nguồn gốc tấn công* (hậu tấn công) [1]. Các giải pháp hiện tại ở bước *phát hiện và phòng chống* tấn công mới chỉ tập trung giải quyết vấn đề phát hiện có luồng lưu lượng tấn công vào hệ thống hay không mà không chỉ ra được các đối tượng gây nên tấn công đó. Các kỹ thuật phát hiện các đối tượng phát tán tấn công thực hiện ở bước hậu tấn công [2][3]. Để có thể vừa phát hiện nguy cơ tấn công đồng thời có thể chỉ ra các đối tượng gây ra nguy cơ đó trong dòng gói tin IP thời gian thực là vấn đề quan trọng đặt ra nhưng chưa có giải pháp, nhằm cảnh báo sớm để có giải pháp ứng phó kịp thời.

Phát hiện sớm các Hot-IP trên mạng và ứng dụng để phát hiện các đối tượng có khả năng là nguy cơ gây nên các cuộc tấn công từ chối dịch vụ, mục tiêu trong các cuộc tấn công này hay phát hiện các máy đang tiến hành quét mạng tìm kiếm lỗ

hổng để phát tán sâu Internet là vấn đề luận án tập trung nghiên cứu. Trong các phương pháp mà luận án đã khảo sát thì phương pháp thử nhóm bất ứng biến là giải pháp thích hợp nhất để triển khai áp dụng.

Bên cạnh đó, các bộ xử lý đa luồng, kỹ thuật xử lý song song, kiến trúc phân tán, đặc điểm của hệ thống mạng tại vị trí triển khai là các yếu tố quan trọng cần xem xét. Đây là những yếu tố có ý nghĩa rất lớn trong việc đưa ra các giải pháp kỹ thuật và có thể kết hợp chúng lại để nâng cao hiệu quả phát hiện Hot-IP và triển khai thực tế.

3. MỤC TIÊU NGHIÊN CỨU

3.1. Mục tiêu tổng quát

Mục tiêu của luận án là xây dựng giải pháp phát hiện các Hot-IP trên mạng máy tính bằng phương pháp thử nhóm bất ứng biến; sử dụng một số kỹ thuật và công cụ toán học kết hợp nhằm nâng cao hiệu quả phát hiện Hot-IP như xây dựng thuật toán và ma trận phân cách phù hợp với vị trí triển khai, xử lý song song, kiến trúc phân tán; áp dụng giải pháp này cho một số bài toán an ninh mạng như phát hiện các đối tượng có khả năng là mục tiêu trong các cuộc tấn công từ chối dịch vụ, phát hiện các đối tượng có khả năng là nguồn phát động tấn công từ chối dịch vụ, các thiết bị có khả năng đang hoạt động bất thường, phát hiện các đối tượng có khả năng là nguồn phát tán sâu Internet và giám sát các Hot-IP trên mạng.

3.2. Các mục tiêu cụ thể

- Nghiên cứu lý thuyết thử nhóm bất ứng biến để đề xuất giải pháp phát hiện các Hot-IP trên mạng.
- Đề xuất phương pháp xây dựng ma trận phân cách tường minh sao cho giảm chi phí tính toán và bộ nhớ khi sử dụng ma trận phân cách.
- Đề xuất cải tiến thuật toán thử nhóm bất ứng biến để giảm thời gian tính toán và phát hiện Hot-IP trên dòng gói tin IP thời gian thực.

- Đề xuất giải pháp kết hợp kỹ thuật xử lý song song, kiến trúc phân tán nhằm phát hiện nhanh các Hot-IP trên mạng.
- Mô hình hóa các bài toán ứng dụng: phát hiện các đối tượng có khả năng là nạn nhân trong các cuộc tấn công từ chối dịch vụ, phát hiện các đối tượng có khả năng là nguồn phát tấn công từ chối dịch vụ, phát hiện các đối tượng có khả năng là nguồn phát tán sâu Internet, phát hiện các thiết bị có khả năng đang hoạt động bất thường về bài toán phát hiện các Hot-IP trên mạng.
- Giám sát các Hot-IP kết hợp với theo dõi tài nguyên hệ thống để điều phối lưu lượng mạng, giảm thiểu các nguy hại trên hệ thống.

4. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU

Nghiên cứu lý thuyết thử nhóm bất ứng biến và áp dụng vào bài toán phát hiện các Hot-IP trên mạng; đồng thời sử dụng kết hợp với kỹ thuật xử lý song song, kiến trúc phân tán để nâng cao hiệu quả của giải pháp phát hiện và cảnh báo sớm các Hot-IP trên mạng.

5. PHƯƠNG PHÁP NGHIÊN CỨU

Nghiên cứu lý thuyết thử nhóm bất ứng biến:

- Hệ thống hóa các khái niệm
- Phân tích và cải tiến các thuật toán trong thử nhóm bất ứng biến

Triển khai thực nghiệm các giải pháp phát hiện Hot-IP:

- Thực nghiệm nhằm xác định các tham số thích hợp
- Phân tích số liệu và tham số thực nghiệm

6. NHỮNG ĐÓNG GÓP CHÍNH CỦA LUẬN ÁN

Sau đây là những đóng góp chính của luận án tập trung vào mục tiêu phát hiện các Hot-IP trên mạng:

(i) **Đề xuất giải pháp phát hiện các Hot-IP trên mạng dựa trên thử nhóm bất ứng biến và một số kỹ thuật kết hợp để nâng cao hiệu quả**

của giải pháp. Trong đó, kỹ thuật tính toán song song được sử dụng trong bước tính vector kết quả của các nhóm thử, sử dụng kiến trúc phân tán trong các hệ thống mạng đa vùng để cảnh báo sớm từ các vùng phát hiện được Hot-IP và lựa chọn kích thước ma trận phù hợp ở vị trí triển khai nhằm giảm áp lực tính toán. Lý thuyết nền tảng cho phương pháp đề xuất là sử dụng phương pháp nối mã để xây dựng tường minh ma trận phân cách. Nhờ đó, không gian lưu trữ được tối ưu thay vì phải lưu trữ toàn bộ ma trận có kích thước lớn trên trường hữu hạn.

(ii) **Đề xuất cải tiến thuật toán thử nhóm bất ứng biến để giảm thời gian tính toán phát hiện các Hot-IP trực tuyến.** Đặc điểm khác biệt của cải tiến là các nhóm thử đến ngưỡng không cần phải cập nhật tiếp tục, danh sách các địa chỉ IP nghi ngờ được xác định và khởi tạo bộ đếm tương ứng, các cập nhật đối với các IP có mặt trong danh sách nghi ngờ được thực hiện thay vì phải cập nhật trong tập tất cả các các bộ đếm của các nhóm thử dựa vào ma trận phân cách.

(iii) **Mô hình hóa 4 bài toán ứng dụng**: (1) phát hiện các đối tượng có khả năng là các nguồn phát tán sâu Internet, (2) phát hiện các thiết bị có khả năng đang hoạt động bất thường, (3) phát hiện các đối tượng có khả năng là mục tiêu hay nguồn phát trong tấn công từ chối dịch vụ về bài toán phát hiện Hot-IP và (4) giám sát hoạt động của các Hot-IP kết hợp với theo dõi tài nguyên mạng để điều phối hay hạn chế hoạt động của các luồng dữ liệu chứa các Hot-IP này. Kỹ thuật được sử dụng là phân tích luồng dữ liệu dựa vào địa chỉ IP nguồn và đích trong các gói tin kết hợp với theo dõi tài nguyên hệ thống làm dữ liệu đầu vào trong thuật toán phát hiện các Hot-IP.

7. GIỚI THIỆU TỔNG QUAN VỀ NỘI DUNG LUẬN ÁN

Nội dung của luận án tập trung vào nghiên cứu phương pháp thử nhóm bất ứng biến và áp dụng vào bài toán phát hiện các Hot-IP trên mạng; đề xuất thuật toán

cải tiến; đề xuất một số kỹ thuật kết hợp để tăng hiệu quả tính toán của giải pháp và đề xuất ứng dụng phát hiện các Hot-IP trong một số bài toán an ninh mạng.

Để giảm không gian lưu trữ và thời gian tính toán, phương pháp nối mã được sử dụng để phát sinh ma trận phân cách tường minh. Phương pháp nối mã cho phép phát sinh ma trận theo từng cột, từ đó sử dụng các tính toán tương ứng mà không cần phải lưu trữ toàn bộ ma trận trong bộ nhớ khi thực thi chương trình. Kết quả này rất quan trọng vì có thể tích hợp vào các bộ định tuyến hay các thiết bị mạng mà không cần tốn nhiều tài nguyên hệ thống. Bên cạnh đó, luận án đề xuất áp dụng kỹ thuật xử lý song song để giảm thời gian kiểm tra nhóm vì trong thử nhóm bất ứng biến thì các nhóm thử được thiết kế trước, kiểm tra một lần, các nhóm thử độc lập nhau và số lượng nhóm thử lớn. Kiến trúc phân tán cũng được đề xuất để triển khai kết hợp nhằm nâng cao hiệu quả trong các hệ thống mạng được tổ chức đa vùng như mạng ở các nhà cung cấp dịch vụ.

Cấu trúc của luận án được tổ chức thành 4 chương. Chương 1 trình bày tổng quan về bài toán Hot-IP, một số khái niệm, khảo sát các nghiên cứu liên quan đến phát hiện tấn công từ chối dịch vụ, phát tán sâu Internet dựa vào các IP trong dòng dữ liệu xuất hiện với tần suất cao, các thuật toán tìm các phần tử tần suất cao trong dòng dữ liệu. Trên cơ sở đó, luận án đề xuất giải pháp phát hiện các Hot-IP trên mạng dùng phương pháp thử nhóm bất ứng biến.

Chương 2 trình bày phương pháp thử nhóm bất ứng biến, một số khái niệm liên quan, phương pháp xây dựng tường minh ma trận phân cách bằng phép nối mã và áp dụng phương pháp thử nhóm bất ứng biến vào bài toán phát hiện các Hot-IP trên mạng. Trong chương này, luận án đề xuất hai thuật toán cải tiến *"Online Hot-IP Detecting"* và *"Online Hot-IP Preventing"* từ phương pháp thử nhóm bất ứng biến để giảm thời gian tính toán và đảm bảo hệ thống hoạt động ổn định, thông suốt khi phát hiện trực tuyến trên cơ sở sử dụng danh sách các địa chỉ IP nghi ngờ. Thuật toán cải tiến còn có khả năng cho kết quả chính xác hơn khi số lượng Hot-IP thực sự lớn hơn số lượng tối đa cho phép của phương pháp thử nhóm bất ứng biến. Bên

cạnh đó, luận án còn trình bày việc thiết lập ngưỡng dựa vào năng lực của hệ thống mạng tại vị trí triển khai để khai thác tối đa khả năng của hệ thống và sử dụng kích thước ma trận phù hợp.

Chương 3 trình bày một số kỹ thuật kết hợp nhằm nâng cao hiệu quả của giải pháp phát hiện các Hot-IP trên mạng. Trong đó, luận án đề xuất kết hợp với kỹ thuật xử lý song song, kiến trúc phân tán trong các hệ thống mạng đa vùng, ý nghĩa và một số căn cứ để lựa chọn các tham số quan trọng trong giải pháp đề xuất áp dụng tại vị trí triển khai cũng được trình bày trong chương này.

Chương 4 trình bày mô hình hóa một số ứng dụng trong lĩnh vực an ninh mạng như phát hiện các đối tượng có khả năng là các nguồn phát hay mục tiêu trong các cuộc tấn công từ chối dịch vụ, phát hiện các thiết bị có khả năng đang hoạt động bất thường trong hệ thống mạng, phát hiện các đối tượng có khả năng là nguồn phát tán sâu Internet về bài toán phát hiện các Hot-IP trên mạng, giám sát các Hot-IP trong một vài chu kỳ thuật toán kết hợp với theo dõi tài nguyên mạng để hạn chế hoạt động của chúng. Việc xác định sớm các đối tượng như đã nêu trên có ý nghĩa quan trọng nhằm giúp các nhà quản trị mạng theo dõi và ứng phó kịp thời, đảm bảo hệ thống hoạt động ổn định, thông suốt.

Trong phần kết luận, luận án tổng kết những kết quả đạt được và bài toán mở cho nghiên cứu tương lai khi áp dụng kết quả luận án vào thực tiễn.

CHƯƠNG 1. TỔNG QUAN VỀ HOT-IP TRÊN MẠNG

1.1. GIỚI THIỆU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như ngày nay, vấn đề an ninh mạng máy tính là một vấn đề quan trọng. Việc đảm bảo an ninh cho hệ thống mạng máy tính cần được xem xét từ cả phía nhà cung cấp dịch vụ cho đến hệ thống mạng nội bộ của các cơ quan, tổ chức, doanh nghiệp. Mục tiêu là để tiến hành các giải pháp phát hiện và ngăn chặn kịp thời các truy cập trái phép vào hệ thống.

Các cuộc tấn công từ chối dịch vụ, đặc biệt là tấn công từ chối dịch vụ phân tán, sự quét mạng để phát hiện lỗ hổng và phát tán sâu trên Internet ngày càng dễ thực hiện nhưng tác hại của nó là đặc biệt nghiêm trọng. Đặc điểm quan trọng trong các cuộc tấn công này là tốc độ cao và thời gian tiến hành rất ngắn. Chính vì nhanh và ngắn như vậy làm cho các nhà quản trị không thể kịp thời chống đỡ, hệ thống mạng bị cạn kiệt tài nguyên, băng thông, dẫn đến tình trạng các dịch vụ mạng bị ngưng trệ không thể đáp ứng tốt cho những người dùng hợp lệ.

Có ba giai đoạn để tiến hành các giải pháp phát hiện và phòng chống tấn công từ chối dịch vụ: (1) *giai đoạn trước khi bị tấn công*, giai đoạn này thực hiện các giải pháp đề phòng như thiết lập các ngưỡng tần suất để phòng tấn công xảy ra; (2) *giai đoạn trong khi bị tấn công*, giai đoạn này sử dụng các kỹ thuật phát hiện và phòng chống với mục tiêu là xác định nhanh có tấn công hay không trong dòng dữ liệu và hành động phản ứng lại tấn công như thế nào; (3) *giai đoạn hậu tấn công*, giai đoạn này sử dụng các kỹ thuật *"dò ngược"* để dò tìm các đối tượng gây ra tấn công [1].

Ở giai đoạn (2), các nghiên cứu về kỹ thuật phát hiện chủ yếu xem xét trong luồng dữ liệu có tiềm tàng khả năng tấn công hay không. Một số giải pháp phát hiện được sử dụng như phân tích thống kê [4][5], phương pháp học máy [6], phương pháp khai phá dữ liệu [7][8], phương pháp dựa vào dấu hiệu được xác định trước [65]. Các phương pháp này có khả năng phát hiện nhanh tấn công trong dòng dữ

liệu nhưng không tìm ra nguồn phát hay xác định nạn nhân trong cuộc tấn công. Phương pháp phát hiện các đối tượng gây nên tấn công thường diễn ra ở bước hậu tấn công.

Các giải pháp phòng chống tấn công từ chối dịch vụ phổ biến hiện nay sử dụng hệ thống phát hiện và phòng chống xâm nhập (IDS/IPS) đặt trước hệ thống máy chủ trong mạng nội bộ để theo dõi, cảnh báo và ngăn chặn [9]. Kỹ thuật được sử dụng trong giải pháp này dựa vào các dấu hiệu được định nghĩa sẵn. Giải pháp này bị hạn chế trong trường hợp xác định tấn công mạng với các dấu hiệu mới.

Trên mạng tốc độ cao như mạng ở phía nhà cung cấp dịch vụ rất cần có giải pháp thực hiện đơn giản và hiệu quả nhằm phát hiện nhanh các đối tượng có khả năng là nguy cơ gây nên các cuộc tấn công này để có thể kịp thời hạn chế ảnh hưởng xấu của chúng. Dựa vào dòng dữ liệu lưu thông qua các thiết bị mạng, các thông tin về địa chỉ nguồn (IP nguồn) và địa chỉ đích (IP đích) xuất hiện với tần suất cao trong một khoảng thời gian rất ngắn (Hot-IP) dẫn đến khả năng các máy chủ có thể đang bị tấn công từ chối dịch vụ, các đối tượng đang phát tán sâu mạng hay các đối tượng đang thực hiện tấn công từ chối dịch vụ. Do đó, việc xác định các đối tượng có khả năng là mục tiêu hay nguồn phát trong tấn công từ chối dịch vụ, các đối tượng có khả năng đang quét mạng để tiến hành phát tán sâu Internet có thể đưa về dạng bài toán phát hiện các Hot-IP trên mạng. Ở đây ta đã sử dụng nhận xét: "Có tấn công dạng từ chối dịch vụ hay phát tán sâu Internet dạng quét không gian địa chỉ IP thì xuất hiện Hot-IP, nhưng Hot-IP chưa chắc bị tấn công". Do đó, luận án nghiên cứu giải pháp dung hòa giữa phòng chống tấn công và tính sẵn sàng của mạng.

Bài toán phát hiện các Hot-IP trên mạng có thể phát biểu đơn giản như sau: cho dòng gói tin IP rất lớn qua mạng, dựa vào tần suất xuất hiện của các IP so với một giá trị ngưỡng tần suất cao định trước. Xác định các Hot-IP trong dòng gói tin IP đó. Việc phát hiện nhanh các Hot-IP chính là phát hiện sớm các đối tượng có khả năng là nguy cơ gây hại trên mạng để tiến hành các giải pháp ứng phó kịp thời. Các

phương pháp phát hiện trên dòng dữ liệu thời gian thực cần phải đơn giản, nhanh chóng và chính xác [10]. Với những mục tiêu như vậy, luận án nghiên cứu và đề xuất giải pháp phát hiện các Hot-IP trên mạng dùng phương pháp thử nhóm bất ứng biến, kết hợp với một số kỹ thuật như xử lý song song và kiến trúc phân tán để nâng cao hiệu quả của giải pháp. Việc áp dụng giải pháp này ở các mạng có số lượng người dùng lớn hay mạng trung gian ở phía nhà cung cấp dịch vụ là phù hợp và hiệu quả. Từ đó, phía nhà cung cấp dịch vụ có thể giúp hạn chế, ngăn chặn và cảnh báo sớm các nguy cơ cho khách hàng của mình, góp phần nâng cao khả năng phục vụ, giảm thiểu ảnh hưởng xấu đến các hoạt động của các dịch vụ, thiết bị trên mạng.

1.2. MỘT SỐ KHÁI NIỆM VÀ ĐỊNH NGHĨA

Khái niệm 1: *Địa chỉ IP (gọi tắt là IP) là chuỗi các ký hiệu dùng để định danh cho các thiết bị trên mạng.*

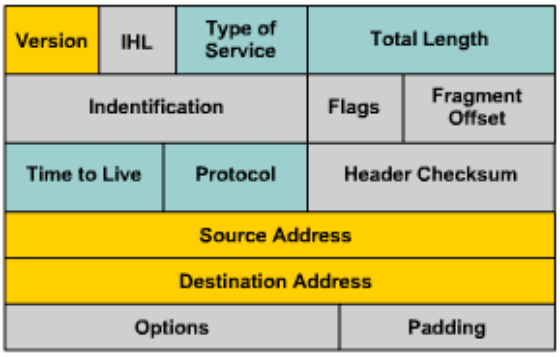
Cấu trúc tổng quát của một địa chỉ IP gồm có hai phần là *Network* và *Host*. Phần *Network* mang giá trị đại diện cho một mạng và phần *Host* đại diện cho các thiết bị trong mạng đó. IPv4 có tổng cộng 32 bit, chia làm 4 phần, mỗi phần ngăn cách nhau bởi dấu ".", được biểu diễn dưới dạng thập phân hoặc nhị phân. Một phiên bản mới hơn là IPv6, địa chỉ IPv6 có tổng cộng 128 bit, chia làm 8 phần, mỗi phần ngăn cách nhau bằng dấu ":", được biểu diễn dưới dạng thập lục phân. Phiên bản mới này đã mở rộng không gian địa chỉ hơn so với IPv4. Tuy nhiên, nó có hạn chế là không tương thích với IPv4. Do đó, vấn đề triển khai gặp khó khăn khi phiên bản IPv4 đang phủ khắp môi trường Internet. Cấu trúc của IPv4-header và IPv6-header được thể hiện ở hình 1.1 và hình 1.2.

Dù các thiết bị trên mạng sử dụng cấu trúc địa chỉ nào cũng không ảnh hướng đến tính tổng quát của giải pháp vì tham số đầu vào của thuật toán là địa chỉ IP được trích ra trong từng gói tin, trong giải pháp xem đó như một giá trị đại diện cho một thiết bị trên mạng mà không sử dụng tới cấu trúc của nó trong thuật toán.

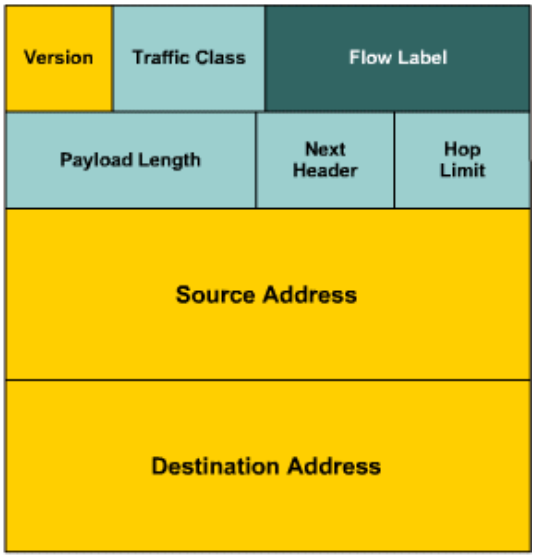
Khái niệm 2: *Gói tin IP là gói tin ở tầng mạng trong mô hình OSI, trong đó có phần IP-header mô tả thông tin ở tầng này. Trong cấu trúc của IP-header chứa*

thông số về địa chỉ IP nguồn và IP đích. Các giá trị địa chỉ này được sử dụng làm tham số đầu vào trong bài toán phát hiện các Hot-IP.

Khái niệm 3: *Dòng gói tin IP là một dãy liên tiếp các gói tin IP* $(a_1, a_2, ..., a_m)$ *luân chuyển trên một đường truyền xác định. Trong đó, mỗi gói tin* a_i *có địa chỉ IP cần phân tích là* s_i *(*s_i *có thể là IP nguồn hay IP đích cần xem xét tùy vào ứng dụng cụ thể).*



Hình 1.1. *Cấu trúc của IPv4-header trong gói tin IPv4*



Hình 1.2. *Cấu trúc của IPv6-header trong gói tin IPv6*

Định nghĩa 1: *Hot-IP trong dòng gói tin IP trên mạng máy tính là những IP xuất hiện với tần suất cao trong khoảng thời gian ngắn xác định trước. Cho dòng gói tin IP có địa chỉ IP tương ứng* $S = (IP_1, IP_2, ..., IP_m)$*, ký hiệu N là số IP khác nhau*

trong m IP thuộc S $(0 \le N \le m)$. *Gọi* $f_i = \left|\left\{ j \middle| IP_i = IP_j; i \ne j; IP_i, IP_j \in S \right\}\right|$, *thì* $\textit{Hot-IP} = \left\{ IP_i \in S \middle| f_i \ge \phi \times m, 0 \le \phi \le 1 \right\}$.

Trong hệ thống mạng ngày nay, tốc độ truyền dữ liệu ngày càng được nâng cao hơn. Các ứng dụng trực tuyến ngày một đa dạng trải dài từ thương mại điện tử đến học tập, giải trí. Từ đó, dòng dữ liệu lưu thông trên mạng ngày một trở nên rất lớn. Phát hiện sớm các đối tượng có khả năng là nguy cơ như các nạn nhân trong cuộc tấn công từ chối dịch vụ, các máy đang quét mạng nhằm phát hiện lỗ hổng để phát tán sâu trên mạng Internet hay một số bất thường khác là vấn đề vô cùng quan trọng. Phát hiện sớm các đối tượng nguy cơ này ở phía nhà cung cấp dịch vụ có ý nghĩa quan trọng nhằm hạn chế, phòng chống, cảnh báo sớm các nguy hại cho hệ thống máy chủ cung cấp dịch vụ của khách hàng. Một trong những đặc trưng cơ bản của chúng là phát tán rất nhanh với một số lượng rất lớn gói tin gửi tới các nạn nhân trong một khoảng thời gian rất ngắn.

Các Hot-IP được phát hiện sớm là cách đơn giản và hiệu quả để xác định các đối tượng có khả năng là nguy cơ gây ra tấn công mạng, phát tán sâu Internet hay một số bất thường như đã đề cập ở trên. Mục tiêu chính của các giải pháp trên dòng dữ liệu là xử lý dòng dữ liệu đầu vào thời gian thực sao cho sử dụng ít không gian lưu trữ và thời gian chạy thuật toán nhanh, các thuật toán chỉ cần tính toán một lần trên dòng dữ liệu đầu vào để cho ra kết quả [10]. Để thực hiện điều này cần phân tích lưu lượng mạng và các gói dữ liệu trong thời gian thực. Các dòng dữ liệu có thể được xem xét ở nhiều mức độ.

- Ở mức độ thứ nhất liên quan đến phân tích *packet log* (nhật ký gói). Mỗi gói tin IP có phần header gồm địa chỉ IP nguồn và IP đích, cổng và các thông tin khác. *Packet log* là một danh sách các thuộc tính trong header của một dãy các gói tin IP gửi qua router.
- Ở mức độ thứ hai liên quan đến phân tích *flow log* (nhật ký dòng). Mỗi *flow* là tập hợp các gói tin có cùng các giá trị của một thuộc tính trong header xác

định nào đó như là IP nguồn hay IP đích. *Flow log* bao gồm thông tin tích lũy về số lượng byte và các gói tin gửi đi, thời gian bắt đầu, thời gian kết thúc và loại giao thức của mỗi luồng dữ liệu đi qua router.

- Ở mức độ thứ ba liên quan đến việc phân tích các *SNMP log*, các gói tin được tập hợp từ các thiết bị được giám sát gửi định thời đến SNMP server.
- Ở mức độ thứ tư liên quan đến dòng dữ liệu thời gian thực, các gói tin trong dòng gói tin được phân tích và xử lý để phát hiện các nguy cơ trong khoảng thời gian rất nhanh.

Luận án tập trung xử lý các dòng dữ liệu ở mức thứ tư.

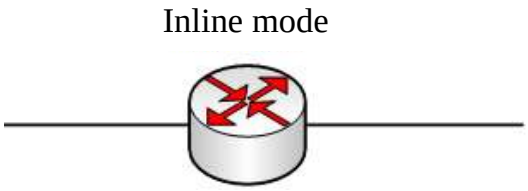
Phần tiếp theo, luận án sẽ trình bày một số vấn đề liên quan đến vị trí thu thập và xử lý dữ liệu; các nghiên cứu liên quan đến bài toán xác định các Hot-IP trên mạng. Trong đó, luận án tập trung phân tích các nghiên cứu về phát hiện tấn công từ chối dịch vụ (DoS và DDoS), phát tán sâu Internet loại *"scanning worm"*; mở rộng các nghiên cứu liên quan về phát hiện các phần tử tần suất cao trong dòng dữ liệu. Từ đó làm cơ sở để lựa chọn giải pháp phù hợp nhất cho bài toán phát hiện các Hot-IP trực tuyến trên mạng.

1.3. VỊ TRÍ THU THẬP VÀ XỬ LÝ DỮ LIỆU

Để thu thập dữ liệu và xử lý trong thuật toán, có hai mô hình cơ bản được sử dụng là mô hình *Inline* và *Promiscuous* [67].

1.3.1. Inline

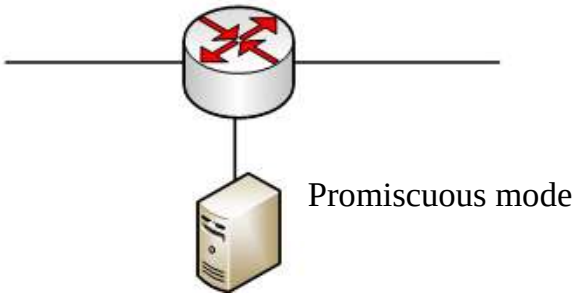
Trong mô hình Inline, thiết bị thu thập và xử lý được đặt trực tiếp trên đường truyền dữ liệu. Ưu điểm của mô hình này là có thể phát hiện, phân tích và ngăn chặn ngay các đối tượng Hot-IP được phát hiện bởi hệ thống. Hạn chế của mô hình này là có thể xảy ra tắc nghẽn với dòng dữ liệu lớn. Hệ thống có tính năng cảnh báo và ngăn chặn như IPS thường sử dụng mô hình này.



Hình 1.3. *Vị trí thu thập dữ liệu dạng Inline*

1.3.2. Promiscuous (passive)

Trong mô hình này, thiết bị thu thập và xử lý nằm song song với đường truyền dữ liệu. Ưu điểm của cách triển khai này là tránh làm tắc nghẽn đường truyền. Hạn chế của nó là hệ thống có thể bị đánh sập trước khi có cảnh báo và ngăn chặn. Ở các hệ thống chỉ có tính năng cảnh báo như IDS thường sử dụng mô hình này.



Hình 1.4. *Vị trí thu thập dữ liệu dạng Promiscuous*

Tham số được sử dụng trong giải pháp của luận án là địa chỉ IP được trích ra trong phần IP-header của các gói tin trong dòng gói tin IP.

1.4. CÁC NGHIÊN CỨU LIÊN QUAN

Địa chỉ IP là địa chỉ đại diện cho thiết bị hoạt động trên mạng. Hot-IP là thuật ngữ được dùng để chỉ địa chỉ IP xuất hiện với tần suất cao trên mạng trong một khoảng thời gian ngắn. Các nghiên cứu liên quan đến Hot-IP chủ yếu được đề cập trong các công trình nghiên cứu về phát hiện và phòng chống tấn công từ chối dịch vụ, các nghiên cứu về một số loại sâu Internet quét không gian địa chỉ để tìm kiếm lỗ hổng và phát tán trên môi trường Internet như loại *"scanning worm"*. Do đó, luận án tập trung phân tích các nghiên cứu liên quan đến hướng này.

Để mở rộng phạm vi so sánh và lựa chọn giải pháp thích hợp, luận án khảo sát các thuật toán phát hiện phần tử tần suất cao trong dòng dữ liệu. Từ đó, luận án có cơ sở lựa chọn giải pháp phù hợp để áp dụng vào bài toán phát hiện các Hot-IP trên mạng.

1.4.1. Các nghiên cứu về tấn công DoS/DDoS

Phát hiện các Hot-IP là phát hiện những IP xuất hiện với tần suất cao trong dòng gói tin IP trong một khoảng thời gian ngắn và là bài toán có ý nghĩa quan trọng. Những Hot-IP này là các đối tượng có khả năng gây nguy cơ tấn công từ chối dịch vụ, có thể là các đối tượng đang tiến hành quét mạng để khai thác lỗ hổng nhằm phát tán sâu Internet hoặc có thể là các mục tiêu trong tấn công từ chối dịch vụ.

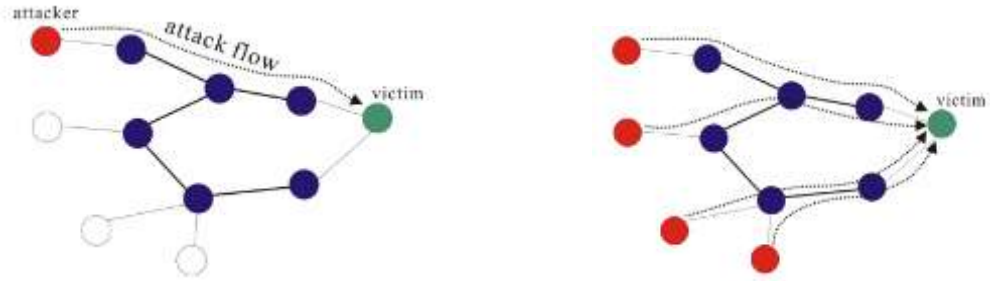
Trong quá trình vận chuyển gói tin qua hệ thống mạng, thiết bị định tuyến đóng vai trò quan trọng. Thiết bị định tuyến hoạt động chính ở tầng mạng *(Layer 3 – Network)* trong mô hình OSI đảm nhận chức năng lựa chọn đường đi cho các gói tin dựa vào địa chỉ IP đích trong các gói tin gửi tới nó và bảng định tuyến được xây dựng trước.

Tần suất xuất hiện cao của các gói tin xuất phát từ một nguồn phát hay đến một nạn nhân nào đó được đề cập chủ yếu trong các nghiên cứu về tấn công từ chối dịch vụ, tấn công của các đối tượng quét mạng nhằm phát tán sâu Internet. Trong đó, các nghiên cứu xác định các đối tượng xuất hiện tần suất cao (các máy phát động tấn công từ chối dịch vụ, các mục tiêu trong tấn công từ chối dịch vụ) chủ yếu được đề cập ở bước hậu tấn công. Các nghiên cứu về phát hiện và phòng chống ở giai đoạn bị tấn công chỉ mới tập trung vào việc kiểm tra luồng dữ liệu có đang bị tấn công hay không [1].

Các nghiên cứu về phát hiện và phòng chống xâm nhập có thể kể đến hai nhóm giải pháp chính: dựa vào dấu hiệu được định nghĩa sẵn và thiết lập ngưỡng tần suất. Giải pháp dựa vào dấu hiệu thực hiện việc so khớp các dấu hiệu được định nghĩa sẵn và thông tin nội dung của các gói tin trong dòng dữ liệu thu thập được.

Việc thiết lập ngưỡng dựa trên các chế độ bình thường được định nghĩa sẵn và sử dụng trong phương pháp thống kê [11], phương pháp học máy [6], khai phá dữ liệu [7][8]. Các phương pháp này gặp khó khăn trong việc định nghĩa các trạng thái bình thường của hệ thống.

Tấn công từ chối dịch vụ, đặc biệt là tấn công từ chối dịch vụ phân tán là dạng tấn công nguy hiểm trên mạng, gây nhiều hậu quả nghiêm trọng và thiệt hại lớn. Mục tiêu của kẻ tấn công là làm tê liệt các ứng dụng, máy chủ, gián đoạn các kết nối, ngăn cản người dùng hợp lệ truy cập vào một dịch vụ nào đó trên mạng. Thông thường trong các cuộc tấn công này, các máy chủ sẽ bị "tràn ngập" bởi hàng loạt các truy vấn trong một khoảng thời gian rất ngắn, dẫn đến quá tải và mất khả năng phục vụ. Tấn công từ chối dịch vụ phân tán hiện nay đã phát triển một cách đáng lo ngại và là mối đe dọa thường trực đối với các hệ thống mạng.



Hình 1.5. *Tấn công DoS và DDoS* [3]

Tấn công DoS và DDoS được minh họa trên hình 1.5. Điểm khác biệt cơ bản giữa tấn công DoS và DDoS là: tấn công DoS xuất phát từ một nguồn còn tấn công DDoS xuất phát từ rất nhiều nguồn phát tấn công, tạo thành một hệ thống mạng lưới gọi là mạng Botnet. Hệ thống Botnet ngày càng lớn về quy mô và trở nên rất nguy hiểm cho bất cứ một hệ thống mạng nào [64]. So với tấn công DoS, tấn công DDoS có sức mạnh lớn hơn rất nhiều lần với số lượng gói tin rất lớn ào ạt gửi tới nạn nhân nhằm chiếm dụng tài nguyên và làm tràn ngập đường truyền của mục tiêu xác định.

Các tấn công này tập trung vào hai mục tiêu chính: (1) gây quá tải cho hệ thống làm cho hệ thống mất khả năng phục vụ người dùng hợp lệ và (2) lợi dụng lỗ

hổng an toàn thông tin của hệ thống, gửi các yêu cầu hoặc các gói tin không hợp lệ làm cho hệ thống sụp đổ.

Đối với loại (1), mỗi hệ thống đều có tài nguyên giới hạn, do vậy khi nhận được quá nhiều yêu cầu dịch vụ giả của kẻ tấn công, hệ thống sẽ sử dụng toàn bộ tài nguyên của mình để đáp ứng các yêu cầu đó, khi đó không còn tài nguyên để đáp ứng các yêu cầu thực sự của người dùng hợp lệ, người dùng hợp lệ sẽ không thể truy cập được vào hệ thống. Đối với loại (2), kẻ tấn công lợi dụng một số lỗ hổng của an toàn thông tin như hoạt động của các giao thức mạng để tấn công. Một số loại tấn công từ chối dịch vụ phổ biến như *"SYN flood"*, *"UDP flood"*, "*ICMP flood"*. Trong các cuộc tấn công DDoS, các nguồn phát tấn công nằm phân tán làm cho việc xác định các kẻ tấn công rất khó khăn.

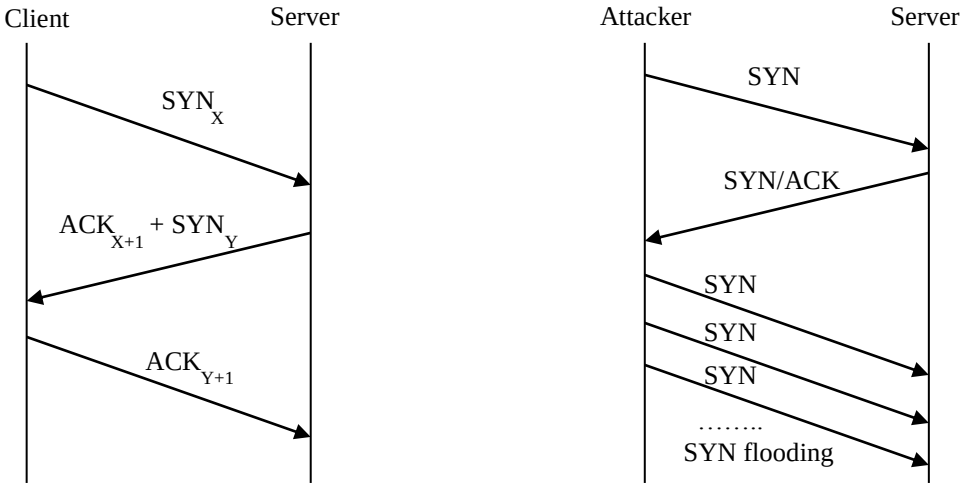
Các giải pháp phát hiện và phòng chống tấn công từ chối dịch vụ được phân làm 4 loại chính thể hiện trong bảng 1.1: *đề phòng*, *phát hiện tấn công*, *phản ứng lại tấn công* và *xác định nguồn phát tấn công* [1]. Trong đó, các nghiên cứu về phát hiện tấn công và phát hiện các nguồn phát là hai vấn đề quan tâm trong luận án này.

Bảng 1.1. *Các giải pháp sử dụng trong các giai đoạn tấn công*

	Trước khi xảy ra tấn công	**Trong khi xảy ra tấn công**	**Sau khi tấn công**
Giải pháp	*Đề phòng*	*Phát hiện và phản ứng lại tấn công*	*Lần theo dấu vết và truy tìm nguồn gốc tấn công*

Có nhiều loại tấn công từ chối dịch vụ, luận án tập trung vào loại làm "tràn ngập" đường truyền và khả năng xử lý của máy chủ. Một số kiểu tấn công làm "tràn ngập" điển hình như "SYN flood", "RST flood", "FIN flood" ở tầng vận chuyển *(Layer 4 – Transport)* và "ICMP flood" ở tầng mạng *(Layer 3- Network)*, tấn công ở tầng ứng dụng [12].

Để hiểu rõ hơn cách khai thác lỗ hổng và tấn công hệ thống, luận án trình bày một kiểu tấn công từ chối dịch vụ cụ thể là tấn công "SYN Flood" gây cho hệ thống máy chủ mất khả năng tiếp nhận các kết nối TCP. Với hoạt động bình thường, các kết nối TCP ở tầng vận chuyển sẽ hoàn thành quá trình bắt tay ba bước được mô tả như hình 1.6(a). Các nguồn tấn công giả mạo địa chỉ IP sẽ không hoàn tất quá trình bắt tay ba bước. Các máy tấn công gửi rất nhiều gói SYN đến máy chủ nạn nhân nhưng không gửi gói ACK để xác nhận và kết thúc quá trình bắt tay ba bước, làm cho máy chủ nạn nhân phải đợi hết thời gian chờ và phải bảo lưu nguồn tài nguyên cho kết nối đó. Nếu quá trình này lặp lại với rất nhiều yêu cầu được gửi đến thì máy chủ nạn nhân không thể xử lý với số lượng lớn các gói tin và sẽ sụp đổ.



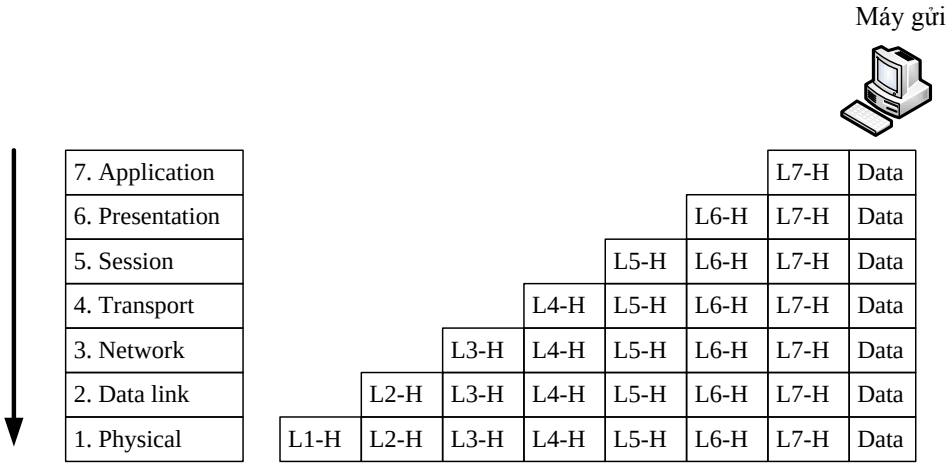
(a) Bắt tay 3 bước *(b) Tấn công TCP SYN*

***Hình 1.6**. Quá trình bắt tay 3 bước và tấn công TCP SYN*

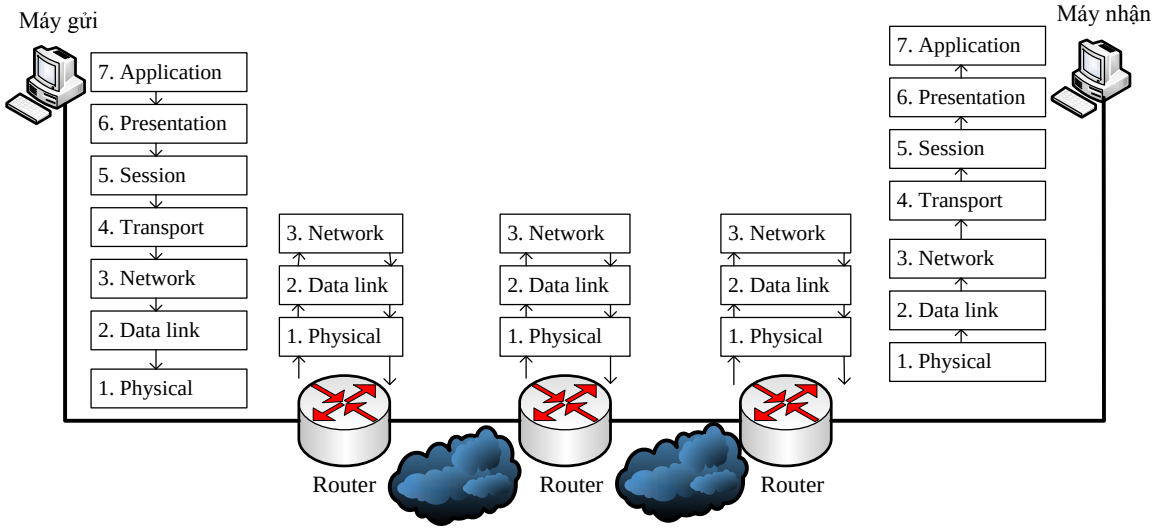
Các gói tin ở tầng vận chuyển hay ở các tầng cao hơn khi gửi đi phải được đóng gói xuống tầng mạng (gắn thêm IP-header, trong đó có chứa thông tin về địa chỉ IP nguồn và IP đích) trước khi gửi đi. Do đó, có thể phát hiện được tần suất xuất hiện của các đối tượng trên mạng chuyển qua môi trường mạng dựa vào thông tin về địa chỉ IP xuất hiện ở tầng mạng. Quá trình đóng gói dữ liệu ở bên máy gửi và quá trình vận chuyển dữ liệu qua mạng được thể hiện trên hình 1.7 và hình 1.8. Như vậy có thể phát hiện các đối tượng có khả năng đang thực hiện tấn công từ chối dịch vụ (thông qua một số lượng lớn các gói tin xuất phát từ một hoặc một số địa chỉ IP

nào đó), các đối tượng có khả năng là mục tiêu trong tấn công từ chối dịch vụ (thông qua một số lượng lớn gói tin gửi đến một hoặc một số địa chỉ IP nào đó) dựa vào bài toán phát hiện các Hot-IP trên mạng.

Nhiều giải pháp phát hiện và phòng chống tấn công từ chối dịch vụ đã được nghiên cứu và đề xuất [1], [2], [4], [6], [8], [9]. Tuy nhiên, cho đến nay vẫn chưa có giải pháp nào có khả năng phòng chống từ chối dịch vụ một cách toàn diện và hiệu quả do tính chất phức tạp, quy mô lớn và khả năng phân tán rất cao của dạng tấn công này. Do vậy, phát hiện sớm các đối tượng có khả năng là nguy cơ tấn công hoặc mục tiêu bị tấn công có vai trò quan trọng trong bài toán an ninh mạng.



Hình 1.7. *Quá trình đóng gói dữ liệu bên máy gửi*



Hình 1.8. *Quá trình vận chuyển dữ liệu qua mạng*

❖ ***Các nghiên cứu về phát hiện tấn công từ chối dịch vụ:***

Tấn công DoS/DDoS làm cạn kiệt tài nguyên và băng thông truy cập đến các máy chủ nạn nhân bằng một số lượng rất lớn các gói tin tấn công được điều khiển gửi đến nạn nhân trong một khoảng thời gian rất ngắn. Mục tiêu quan trọng nhất của các cơ chế phát hiện DoS/DDoS là phát hiện chúng càng sớm càng tốt và tiến hành các giải pháp ngăn chặn càng gần nguồn phát tấn công càng tốt. Các giải pháp phát hiện tấn công từ chối dịch vụ gồm phương pháp phân tích thống kê, phương pháp khai phá dữ liệu, phương pháp học máy, phương pháp dựa vào dấu hiệu đã được định nghĩa sẵn.

Phát hiện tấn công sử dụng phương pháp phân tích thống kê được trình bày trong [13], [15]. Trong phương pháp này, dữ liệu đầu vào được thu thập và ước lượng tần suất xuất hiện dựa vào các đặc trưng luồng lưu lượng để phát hiện có tấn công hay không.

Phát hiện tấn công dùng phương pháp học máy được đề cập trong nghiên cứu [6]. Phát hiện tấn công dùng phương pháp khai phá dữ liệu được trình bày trong [7], phát hiện tấn công sử dụng phương pháp phân tích entropy [11][17], sử dụng mạng nơron nhân tạo để phát hiện tấn công được trình bày trong [18].

Các giải pháp đang triển khai hiện tại trên các thiết bị tường lửa, IDS/IPS để phát hiện, phòng chống tấn công DoS/DDoS gặp phải khó khăn trong việc xác định trạng thái bình thường để đặt ngưỡng và cho kết quả có độ chính xác không cao. Trong các giải pháp này, điểm mạnh là dựa vào các dấu hiệu đã được định nghĩa trước.

❖ ***Dò tìm các nguồn phát tấn công:***

Khi phát hiện có tấn công DoS/DDoS xảy ra và ngắt hệ thống nạn nhân ra khỏi các tài nguyên, việc truy tìm nguồn gốc tấn công được tiến hành. Phương pháp "dò ngược" để phát hiện các kẻ tấn công từ chối dịch vụ được đề cập trong các nghiên cứu [2], [3]. Kỹ thuật "dò ngược" xuất phát từ thiết bị định tuyến gần nhất với máy chủ nạn nhân và kiểm tra tương tác với các thiết bị định tuyến chiều

“upstream” để xác định các kẻ tấn công. Hạn chế của các giải pháp này là phải cài đặt dấu hiệu nhận diện ở các router trên đường đi của gói tin hoặc gắn thêm thông tin ghi dấu đường đi trong các gói tin. Điều này chỉ có thể thực hiện trong mạng nội bộ, khó thực thi trong môi trường Internet vì không thể can thiệp vào tất cả các thiết bị định tuyến trên môi trường Internet. Đây là giải pháp được sử dụng ở giai đoạn hậu tấn công.

Một vài phương pháp phát hiện kẻ tấn công được đề xuất trong tấn công dịch vụ Web như phương pháp sử dụng CAPTCHA. Hiện tại cơ chế CAPTCHA gần như là cơ chế bảo mật hiệu quả để chống lại các kẻ tấn công DoS/DDoS [19]. Nhược điểm của nó là làm khó chịu khi người dùng bị gián đoạn bởi CAPTCHA và không có tác dụng trong các loại tấn công làm “tràn ngập” gây tê liệt hoạt động của máy chủ. Trong các loại tấn công làm “tràn ngập” thì giải pháp sử dụng là phát hiện luồng tấn công và phát hiện các đối tượng tấn công dựa vào phương pháp “dò ngược”, sử dụng kỹ thuật hạn chế tốc độ để hạn chế các tấn công này.

Hầu hết các nghiên cứu về giải pháp phát hiện tấn công từ chối dịch vụ ở giai đoạn tấn công chủ yếu tập trung vào việc phát hiện có tấn công hay không hơn là xác định các kẻ tấn công [1]. Phương pháp sử dụng trong bài toán phát hiện tấn công là định thời so sánh trạng thái hiện tại của hệ thống với mô hình hệ thống bình thường được thiết lập trước, từ đó phát hiện lưu lượng tấn công. Ưu điểm của phương pháp này là đơn giản và phát hiện rất nhanh các tấn công xuất hiện, tuy nhiên không thể cung cấp thông tin về địa chỉ của các kẻ tấn công. Do đó để phát hiện nguồn phát tấn công phải dựa vào cơ chế “dò ngược” ở bước hậu tấn công.

Có ba vị trí triển khai giải pháp phát hiện và phòng chống tấn công từ chối dịch vụ: *phía mạng của các máy chủ nạn nhân, vị trí mạng trung gian, vị trí mạng nguồn phát tấn công* [1]. Trong các mạng trung gian như mạng ở các nhà cung cấp dịch vụ, việc phát hiện các đối tượng có khả năng là mục tiêu trong các cuộc tấn công từ chối dịch vụ dựa vào phân tích lưu lượng đi qua nó có ý nghĩa quan trọng. Từ việc phát hiện này có thể giúp cảnh báo sớm cho khách hàng để tiến hành các

giải pháp ứng phó kịp thời hoặc loại bỏ các nguy cơ này để đảm bảo hệ thống hoạt động ổn định, thông suốt.

Phương pháp thử nhóm bất ứng biến có thể xác định các đối tượng có khả năng gây tấn công và các đối tượng có khả năng là mục tiêu trong tấn công từ chối dịch ngay ở giai đoạn phát hiện tấn công. Đồng thời phương pháp thử nhóm bất ứng biến cho kết quả tốt ở khía cạnh thời gian, độ chính xác cao và mức độ đơn giản của giải pháp. Luận án sẽ trình bày một số nghiên cứu liên quan về phương pháp thử nhóm bất ứng biến trong dòng dữ liệu phát hiện phần tử tần suất cao để thấy được những ưu điểm của phương pháp này so sánh với các phương pháp khác và khả năng áp dụng vào bài toán phát hiện các Hot-IP trực tuyến trên mạng.

1.4.2. Các nghiên cứu về sâu Internet

Sâu máy tính là chương trình máy tính có khả năng tự nhân bản và phát tán đến các thiết bị trên mạng bằng cách khai thác các lỗ hổng của các thiết bị này. Sâu máy tính chia làm 2 loại: sâu mạng (*network worm*) và không phải sâu mạng (*non-network worm*). Sâu mạng có thể phát tán bằng cách khai thác các lỗ hổng của các dịch vụ mạng. Sâu mạng được chia làm 2 loại: "*scanning worm"* và *"non-scanning worm"*. *"Scanning worm"* tìm các thiết bị trên mạng có các lỗ hổng dịch vụ mạng bằng cách quét không gian địa chỉ và cổng dịch vụ. Trong các loại sâu *"scanning worm", "routing worm"* và *"hit-list worm"* là những sâu nguy hiểm, phát tán dựa vào thông tin trong bảng định tuyến và danh sách địa chỉ IP (hit-list) với tốc độ cao. Sâu Internet khai thác lỗ hổng của các thiết bị trên môi trường Internet để tiến hành phát tán và lây nhiễm.

	Giai đoạn trước khi phát tán sâu	**Giai đoạn phát tán sâu → cách ly**	**Giai đoạn sau khi nhiễm sâu**
Giải pháp	Đề phòng phát tán sâu	Hạn chế sâu lây lan Phát hiện sâu	Diệt sâu

Hình 1.9. *Các giai đoạn phát tán và giải pháp phòng chống sâu mạng*

Hình 1.9 mô tả các giải pháp phòng chống sâu ở từng giai đoạn hoạt động của chúng [21]. Trong các giai đoạn này, luận án xem xét một số nghiên cứu về các kỹ thuật trong giai đoạn phát tán sâu Internet. Mục tiêu của giải pháp là phát hiện sự tồn tại của sâu càng nhanh càng tốt bằng cách phân tích lưu lượng trên mạng.

Loại sâu *"routing worm"* khai thác bảng định tuyến để quét mạng dựa trên thông tin định tuyến BGP [20]. Trong đó, "hit-list worm" không chỉ lan truyền nhanh hơn mà còn có thể tiến hành các cuộc tấn công đến các quốc gia xác định, các công ty, ISP hay các AS. So với các loại sâu khác, *"routing worm"* có thể gây ra tình trạng tắc nghẽn cho hệ thống đường trục Internet và gây khó khăn trong việc phát hiện.

Hoạt động lây nhiễm sâu gồm các giai đoạn sau: *phát hiện mục tiêu*, *truyền sâu*, *kích hoạt* và *lây nhiễm* [21], [22], [23]. Quá trình hoạt động lây nhiễm sâu Internet ở hai giai đoạn đầu (*phát hiện mục tiêu* và *truyền sâu*) ảnh hưởng đến hoạt động của mạng, nên các hành vi của chúng ở hai giai đoạn này rất quan trọng để tiến hành triển khai các giải pháp phát hiện. Một số đặc điểm quan trọng cần lưu ý để tạo thuận lợi cho việc phát hiện chúng là: ở bước phát hiện mục tiêu, phương pháp đơn giản nhất các sâu hay sử dụng là *"quét mù"*, có 3 phương pháp được sử dụng trong *"quét mù"* là: *quét tuần tự*, *quét ngẫu nhiên* và *quét kết hợp*. Phương pháp này có tính cơ hội và tỷ lệ thất bại cao. Phiên bản nâng cấp từ phương pháp quét này là "routing worm". "Routing worm" là một bước cải tiến với không gian quét nhỏ hơn. "Routing worm" sử dụng thông tin được cung cấp bởi bảng định tuyến BGP để thu hẹp phổ quét và các hệ thống mục tiêu cụ thể trong một vị trí địa lý như một quốc gia, một nhà cung cấp dịch vụ (ISP), hoặc một hệ thống mạng tự trị (AS). "Routing worm" có khả năng lây lan nhanh hơn sâu truyền thống gấp nhiều lần [24], [28].

Sau bước phát hiện mục tiêu, sâu được nhân bản và gửi đến mục tiêu. Có nhiều mô hình phát tán sâu, theo [28] có ba mô hình phát tán sâu: *self-carried*, *second channel* và *embedded*. *"Self-carried"* thực hiện đơn giản, phần *worm-*

payload được truyền trong gói tin của chính nó. Một số loại sâu khác truyền qua *"second channel"* nghĩa là truyền qua một kênh khác, sau khi tìm thấy mục tiêu, sâu sẽ đến mục tiêu, *worm-payload* từ Internet hoặc một máy đã nhiễm trước đó thông qua *"backdoor"* được cài đặt sử dụng RPC hoặc các ứng dụng khác. Phương pháp *"embedded"* hoạt động rất thận trọng, rất khó phát hiện. Bên cạnh 3 phương pháp trên, botnet cũng được sử dụng để truyền sâu, thư rác và thực hiện tấn công từ chối dịch vụ phân tán [25]. Hai giao thức sử dụng ở bước truyền sâu là TCP và UDP.

Các thuật toán phát hiện có thể chia thành 2 dạng: *dựa vào dấu hiệu* và *dựa vào sự bất thường* [24], [26]:

(1) Phát hiện sâu dựa vào dấu hiệu được định nghĩa sẵn (signature-based): phát hiện dựa vào dấu hiệu là kỹ thuật truyền thống được sử dụng cho hệ thống phát hiện xâm nhập (IDS) và thường sử dụng cho phát hiện các cuộc tấn công đã biết.

Loại này không quan tâm tới làm thế nào để tìm các mục tiêu. Trong đó, các dấu hiệu được định nghĩa sẵn so khớp với *payload* (nơi chứa mã sâu thực sự) để phát hiện sâu. Giải pháp này bị hạn chế trong trường hợp các sâu được thay đổi *payload* để tránh hệ thống phát hiện. Ngoài thay đổi cách thể hiện của nó trong payload, sâu còn có thể thay đổi hành vi. Nếu sâu sử dụng mô hình mã hóa phức tạp để che dấu mục đích thực sự của nó thì càng gây khó khăn hơn trong việc phòng chống.

(2) Phát hiện sâu dựa vào sự bất thường (anomaly-based)*:* Các dấu hiệu của một sâu mới xuất hiện là chưa biết. Tất cả các sâu phát tán rộng được biết đến cho đến nay thông thường tạo ra lưu lượng dữ liệu lớn và hầu hết là sử dụng cơ chế "quét mù" để tìm kiếm lỗ hổng của các thiết bị trên mạng và lây nhiễm [21]. Do đó có rất nhiều địa chỉ mục tiêu không tồn tại, từ đó xuất hiện dấu hiệu bất thường.

Hệ thống phát hiện sâu dựa vào sự bất thường không dựa vào *payload* như dạng (1) mà dựa vào *header* của gói tin. Có 3 mục đích phổ biến cho các gói tin được gửi hoặc nhận của một thiết bị là: *khởi tạo kết nối, chỉ ra thất bại trong nỗ lực*

kết nối và *gửi dữ liệu qua một kết nối đã thiết lập*. Hệ thống cũng có thể tiếp tục theo dõi các dữ liệu giữa địa chỉ nguồn và đích để tìm ra các đối tượng nào đang quét mạng.

Vấn đề thách thức lớn đối với giải pháp phát hiện dựa vào sự bất thường là dựa trên việc định nghĩa những hành vi mạng bình thường là gì, quyết định các ngưỡng để kích hoạt báo động. Nếu các mô hình bình thường không được định nghĩa một cách cẩn thận sẽ có rất nhiều cảnh báo sai.

Như vậy, qua các phân tích về hai lĩnh vực nghiên cứu liên quan là phát hiện các đối tượng và mục tiêu trong tấn công từ chối dịch vụ và vấn đề phát tán sâu Internet đối với một số loại sâu như "scanning-worm" liên quan đến nghiên cứu các IP tần suất cao trên mạng cho thấy các giải pháp hiện tại tập trung vào việc phát hiện có tồn tại tấn công hay không trong bước phát hiện và phòng chống tấn công. Việc xác định các đối tượng gây ra tấn công được thực hiện ở bước hậu tấn công.

Do vậy, cần một giải pháp có thể cân bằng điều này, nghĩa là có thể nhanh chóng phát hiện các nguy cơ và đồng thời chỉ ra được các đối tượng này là những IP nào ở giai đoạn xảy ra tấn công (trong tấn công từ chối dịch vụ) hay ở giai đoạn phát tán sâu Internet. Giải pháp đặt ra được đưa về giải bài toán phát hiện Hot-IP trên mạng mà luận án nghiên cứu giải quyết. Vấn đề này có ý nghĩa quan trọng trong các mạng trung gian ở ISP hoặc các hệ thống cung cấp dịch vụ trên Internet được tổ chức dạng đa vùng nhằm hỗ trợ để tiến hành giám sát, hạn chế sớm các nguy hại, đảm bảo sự hoạt động ổn định trên mạng và cảnh báo sớm cho khách hàng.

1.4.3. Các nghiên cứu về thuật toán phát hiện phần tử tần suất cao

Dòng dữ liệu là một dãy tuần tự các phần tử lưu thông trên mạng có đặc điểm là tốc độ truyền dữ liệu nhanh, gây ra nhiều khó khăn trong việc truyền và tính toán của các giải pháp liên quan đến các dữ liệu này. Các thuật toán trên dòng dữ liệu luôn xem xét các yếu tố liên quan nhằm giảm không gian lưu trữ, thời gian tính toán của chương trình [27], [28].

Các phần tử tần suất cao trong dòng dữ liệu có thể đại diện cho nhiều đối tượng khác nhau trên mạng. Các phần tử trong dòng dữ liệu có thể đại diện cho các câu truy vấn gửi đến một *Internet Search Engine*, lúc này các phần tử tần suất cao là các từ khóa tìm kiếm thông dụng. Đối với Web Proxy các phần tử có thể là các URL được các máy trong mạng gửi đến và các phần tử tần suất cao là các URL được yêu cầu nhiều. Các thiết bị định tuyến trên Internet kết nối với nhau để truyền các gói tin IP đến đích, dữ liệu qua nó rất lớn. Do đó, trong việc quản lý một hệ thống mạng lớn, người quản trị cần phát hiện nhanh các đối tượng hoạt động bất thường trên mạng, các đối tượng này có thể là nguyên nhân gây ra các sự cố như các tấn công từ chối dịch vụ, các mục tiêu trong tấn công từ chối dịch vụ, các máy phát tán sâu mạng, các thiết bị hoạt động bất thường trong hệ thống như đã trình bày ở phần trước.

Vì mức độ quan trọng và khả năng ứng dụng rộng rãi của bài toán tìm các phần tử tần suất cao trong dòng dữ liệu nên đã có rất nhiều thuật toán được đề xuất để giải quyết bài toán này. Các thuật toán tiêu biểu tìm các phần tử tần suất cao trong dòng dữ liệu có thể kể đến là: thuật toán *Majority* được đề xuất bởi Moore năm 1982 [30], thuật toán *Frequent* được đề xuất bởi Misra và Gries năm 1982 [31], thuật toán *LossyCounting* được đề xuất bởi Manku và Motwani năm 2002 [32], thuật toán *SpaceSaving* được đề xuất vào năm 2005 bởi Metwally và các cộng sự [33], thuật toán *CountSketch* được đề xuất bởi Charika cùng các cộng sự vào năm 2002 [34], thuật toán *CountMin* của hai tác giả Cormode và Muthukrishnan năm 2005 [28]. Ngoài các thuật toán trên, một giải pháp khác được sử dụng để phát hiện các phần tử tần suất cao là phương pháp thử nhóm. Đây là giải pháp có nhiều ưu điểm và có khả năng phát triển để phát hiện các Hot-IP trực tuyến trên mạng.

Các thuật toán tìm phần tử tần suất cao trong dòng dữ liệu kể trên được chia thành hai nhóm chính: các thuật toán *"counter-based"* và các thuật toán *"Sketch"* được mô tả ở bảng 1.2. Các thuật toán loại *"counter-based"* giám sát một tập các phần tử từ dòng dữ liệu đầu vào cùng với một biến đếm tương ứng với mỗi phần tử được giám sát, sau đó một tập các luật tương ứng cho mỗi thuật toán sẽ được áp

dụng trên danh sách các phần tử này để tìm ra các phần tử tần suất cao cũng như tần suất xuất hiện ước lượng của các phần tử. Các thuật toán loại *"Sketch"* không giám sát một tập các phần tử từ dòng dữ liệu mà xem dòng dữ liệu đầu vào như một vector với mỗi tọa độ của vector là tần suất xuất hiện của một phần tử tương ứng trong dòng dữ liệu, dựa trên các tần số ước lượng này sẽ tính toán ra các phần tử tần suất cao trong dòng dữ liệu.

Bảng 1.2. *Phân nhóm các phương pháp tìm phần tử tần suất cao*

	Counter-based	**Sketch**
Phương pháp giải quyết	• Giám sát một tập các phần tử từ dòng dữ liệu cùng với biến đếm tương ứng • Một tập các luật sẽ được áp dụng trên danh sách các phần tử này để tìm ra phần tử tần suất cao	• Xem dòng dữ liệu như một vector, với mỗi tọa độ là tần suất xuất hiện của mỗi phần tử tương ứng • Dựa vào các tần suất ước lượng này, tính toán ra các phần tử tần suất cao
Thuật toán	Majority, Frequent, LossyCounting, SpaceSaving	Count-Sketch, Count-Min Sketch

❖ ***Các thuật toán loại "counter-based":***

Thuật toán Majority được đề xuất bởi nhóm của Boyer-Moore [30] và nhóm Fischer-Salzberg năm 1982 [35] dùng để tìm phần tử tần suất cao trong dòng dữ liệu. Phần tử tần suất cao được xác định ở đây là phần tử có số lần xuất hiện nhiều hơn nửa trong tổng số các gói trong dòng dữ liệu.

Bài toán được phát biểu như sau: cho dòng dữ liệu $\sigma = \langle a_1, a_2, ..., a_m \rangle$, trong đó có *N* phần tử phân biệt và vector tần suất $f = (f_1, ..., f_N)$, với $f_1 + \cdots + f_N = m$. Vấn đề phần tử tần suất cao được xác định như sau: nếu $\exists j : f_j > m/2$, thì xuất kết quả *j*.

Ý tưởng cơ bản của thuật toán Boyer-Moore-Fischer-Salzberg như sau: chúng ta sẽ lưu trữ một biến đếm *counter* và một phần tử, gọi là phần tử hiện hành, gán *counter =1* và phần tử đầu tiên trong dòng dữ liệu là phần tử hiện hành, với mỗi phần tử mới *a* trong dòng dữ liệu (từ phần tử thứ 2 trở đi). Nếu *a* giống phần tử hiện hành thì tăng *counter* lên 1. Trường hợp nếu *a* khác phần tử hiện hành thì: nếu *counter = 0* thì thay phần tử hiện hành bằng *a* và gán *counter = 1*; nếu *counter > 0* thì giảm *counter* đi 1.

Phương pháp này về bản chất được thực hiện qua 2 pha. Pha thứ nhất tìm ra phần tử tần suất cao nếu nó tồn tại (xác định các ứng cử viên là tần suất cao). Pha thứ 2 là kiểm tra tần suất xuất hiện của chúng, xác định xem phần tử tìm được trong pha thứ nhất có đúng thật là phần tử tần suất cao hay không. Khi kết thúc, nếu có tồn tại một phần tử tần suất cao (xuất hiện hơn *m/2* lần) thì phần tử tần suất cao chính là phần tử hiện hành. Rất tiếc là nếu không có phần tử tần suất cao thì thuật toán này không phân biệt được. Các thuật toán qua 2 pha như thế này không thích hợp cho bài toán trên dòng dữ liệu thời gian thực vì bài toán trên dòng dữ liệu yêu cầu các thuật toán xử lý qua dòng dữ liệu một lần và cho ra kết quả.

Đối với thuật toán *Majority* chỉ có một phần tử tần suất cao được tìm thấy sau khi thuật toán kết thúc. Một số thuật toán sau này đều có thể tìm được tất cả các phần tử có tần suất xuất hiện lớn hơn một tỉ lệ phần trăm của tổng số phần tử trong dòng dữ liệu tính đến thời điểm hiện tại. Thuật toán *Frequent* về bản chất là sự tổng quát hóa của thuật toán *Majority*, chỉ khác là thay vì giám sát một biến đếm và một phần tử như thuật toán *Majority* thì thuật toán *Frequent* giám sát *k* phần tử và các biến đếm tương ứng với mỗi phần tử.

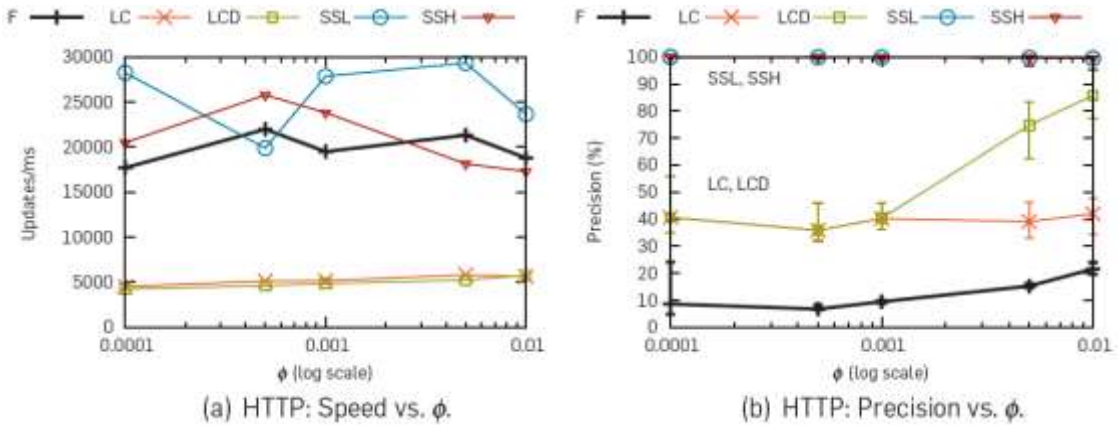
Thuật toán *Frequent* được đề xuất bởi Misra và Gries năm 1982 [31] để tìm tất cả các phần tử có tần suất xuất hiện lớn hơn *m/k*. Thuật toán sử dụng *k-1*cặp (item, counter), với mỗi phần tử mới đến thì thuật toán thực hiện các tác vụ sau: tăng bộ đếm tương ứng nếu phần tử này đã tồn tại. Nếu số phần tử đã được lưu nhỏ hơn *k* thì lưu phần tử mới với biến đếm tương ứng đặt bằng 1; ngược lại, giảm tất cả các bộ đếm. Nếu có bất kỳ bộ đếm nào bằng 0 thì xóa phần tử tương ứng. Misra và Gries cài đặt sử dụng tìm kiếm trên cây cân bằng. Thuật toán này cũng không thích hợp cho các ứng dụng trên dòng dữ liệu vì phải trải qua 2 pha mới kết luận được kết quả của bài toán.

Thuật toán *LossyCounting* được đề xuất bởi Manku và Motwani năm 2002 [32]. Thuật toán này sử dụng cấu trúc lưu trữ các bộ gồm: phần tử (*item*), chặn dưới của bộ đếm và giá trị *delta* là giá trị khác nhau giữa chặn trên và chặn dưới. Khi xử lý phần tử thứ *i* trong dòng dữ liệu, nếu phần tử đã được lưu trữ thì tăng chặn dưới tương ứng của nó thêm 1, ngược lại tạo một bộ mới cho phần tử này với chặn dưới đặt bằng 1 và $delta = \lfloor i/k \rfloor$. Các phần tử sẽ bị xóa nếu có chặn trên nhỏ hơn $\lfloor i/k \rfloor$. Kết quả trả về là các phần tử được lưu có tần suất lớn hơn m/k.

Thuật toán *SpaceSaving* được đề xuất bởi Metwally và cộng sự năm 2005 [33]. Thuật toán này sử dụng cấu trúc dữ liệu gồm lưu trữ *k* bộ *(item, count)*, khởi tạo với *k* phần tử phân biệt và các bộ đếm tương ứng của nó. Nếu một phần tử mới đến chưa được lưu trữ, thay thế cho phần tử có giá trị đếm nhỏ nhất và khởi tạo bộ đếm bằng giá trị đếm nhỏ nhất cộng thêm 1.

Trong công trình nghiên cứu của Cormode và Hadjieleftheriou [10] đã thực nghiệm so sánh các thuật toán này. Nhóm tác giả này thực hiện với 10.000.000 gói tin HTTP với ngưỡng tần suất ϕ từ 0,0001 đến 0,01. Các thuật toán "counter-based" trên hình 1.10(a) thể hiện số lượng gói tin cập nhật trên phần nghìn giây, hình 1.10(b) thể hiện mức độ chính xác của thuật toán. Hình 1.10(a) cho biết số lượng gói tin cập nhật trên phần nghìn giây của các thuật toán *SpaceSaving* và *Frequent*

nhanh hơn *LossyCounting*. Nhìn chung, thuật toán *SpaceSaving* cho kết quả tốt hơn các thuật toán khác trong nhóm các thuật toán *"Counter-based"*.



F: Frequent; **LCD**: LossyCounting; **SSH**: SpaceSaving sử dụng cấu trúc heap; **SSL**: LossyCounting sử dụng danh sách liên kết.

Hình 1.10. *So sánh các thuật toán loại "counter-based"* [10].

Các thuật toán *"counter-based"* lưu trữ mỗi đối tượng bằng một bộ đếm nên tốn nhiều không gian lưu trữ với số lượng rất lớn các đối tượng trên mạng, đặc biệt trên mạng ở các nhà cung cấp dịch vụ, không thích hợp cho bài toán phát hiện các Hot-IP được thiết lập trên môi trường mạng với các thiết bị có tài nguyên hạn chế.

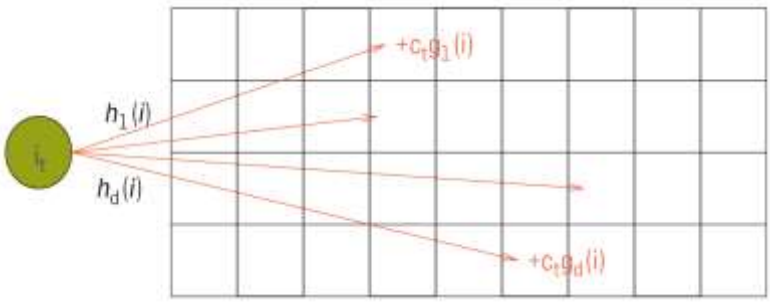
❖ ***Các thuật toán loại "Sketch":***

Không giống như các thuật toán "*Counter-based"*, các thuật toán dựa trên "*Sketch"* không giám sát một tập các biến đếm mà các thuật toán này ước lượng tần suất của các phần tử. Tần suất ước lượng của các phần tử được tính thông qua một tập các mảng của các biến đếm. Mỗi phần tử được đưa vào các biến đếm thông qua các hàm băm, các mảng biến đếm này được cập nhật mỗi khi có phần tử được đưa đến.

Thuật ngữ *"sketch"* được dùng để chỉ một cấu trúc dữ liệu, một ánh xạ tuyến tính cho vector tần suất đầu vào. Các thuật toán dựa vào *sketch* sử dụng các hàm băm để xác định các ánh xạ tuyến tính. Hai thuật toán tiêu biểu cho loại này là *Count-Sketch* và *Count-Min*.

Thuật toán *Count-Sketch* được đề xuất bởi Charikar và các cộng sự vào năm 2002, sử dụng cấu trúc dữ liệu *sketch* để giải quyết bài toán tìm *k* phần tử phổ biến trong dòng dữ liệu [34]. Thuật toán nhận vào dòng dữ liệu *S*, một số nguyên *k* và một số thực ε, sau đó tính toán và trả về *k* phần tử với đảm bảo rằng: mọi phần tử trong *k* phần tử được trả về đều có tần suất lớn hơn $(1-\varepsilon)n_k$. Trong đó, n_k là tần suất của phần tử thứ *k* trong dòng dữ liệu.

Sketch là một mảng C (2 chiều) với d dòng và w bộ đếm, sử dụng 2 hàm băm cho mỗi dòng: hàm h_j ánh xạ các phần tử vào các bộ đếm [w] và hàm g_j ánh xạ các phần tử vào {-1, +1}. Với mỗi phần tử *i* đầu vào thực hiện cộng $g_j(i)$ vào $C[j,h_j(i)]$ trong dòng *j* (với $1 \le j \le d$). Hình 1.11 mô tả cấu trúc dữ liệu *sketch*.



Hình 1.11. *Cấu trúc dữ liệu sketch* [28]

Thuật toán *Count-Min* được đề xuất bởi Cormode và Muthukrishnan năm 2005 [28], trong thuật toán này sử dụng cấu trúc dữ liệu *sketch* là mảng 2 chiều gồm *d* dòng và *w* bộ đếm, sử dụng *d* hàm băm h_j, mỗi dòng sử dụng một hàm băm. Mỗi cập nhật ánh xạ vào *d* thành phần trong dãy, khi đó mỗi phần tử được tăng lên. Ước lượng tần suất cho mỗi phần tử là $f_i^* = \min_{1 \le j \le d} C[j, h_j(i)]$.

Thuật toán tìm phần tử tần suất cao dựa trên *Count-Min* và *Count-Sketch* sử dụng cấu trúc dữ liệu giống nhau để ước lượng tần suất của các phần tử trong dòng dữ liệu. Điểm khác nhau chính của hai thuật toán này là các ước lượng tần suất, thuật toán *Count-Sketch* lấy trung vị của biến đếm, thuật toán *Count-Min* lấy giá trị nhỏ nhất của các biến đếm.

Như vậy, một cách tổng thể có thể thấy rằng các thuật toán *Count-Min* và *Count-Sketch* có nhiều lợi thế để sử dụng cho các bài toán trên dòng dữ liệu lớn vì việc sử dụng các hàm băm sẽ không cần sử dụng từng bộ đếm cho mỗi đối tượng cần xử lý, từ đó sẽ ít tốn không gian lưu trữ trong quá trình tính toán của chương trình.

Nhóm nghiên cứu của Cormode sử dụng phương pháp thử nhóm bất ứng biến để phát hiện các phần tử tần suất cao trong dòng dữ liệu [27]. Trong đó, cấu trúc dữ liệu được phát triển từ phương pháp *Count-Min*. Phương pháp thử nhóm bất ứng biến sử dụng để xác định các phần tử tần suất cao được trình bày sau đây.

1.4.4. Phương pháp thử nhóm

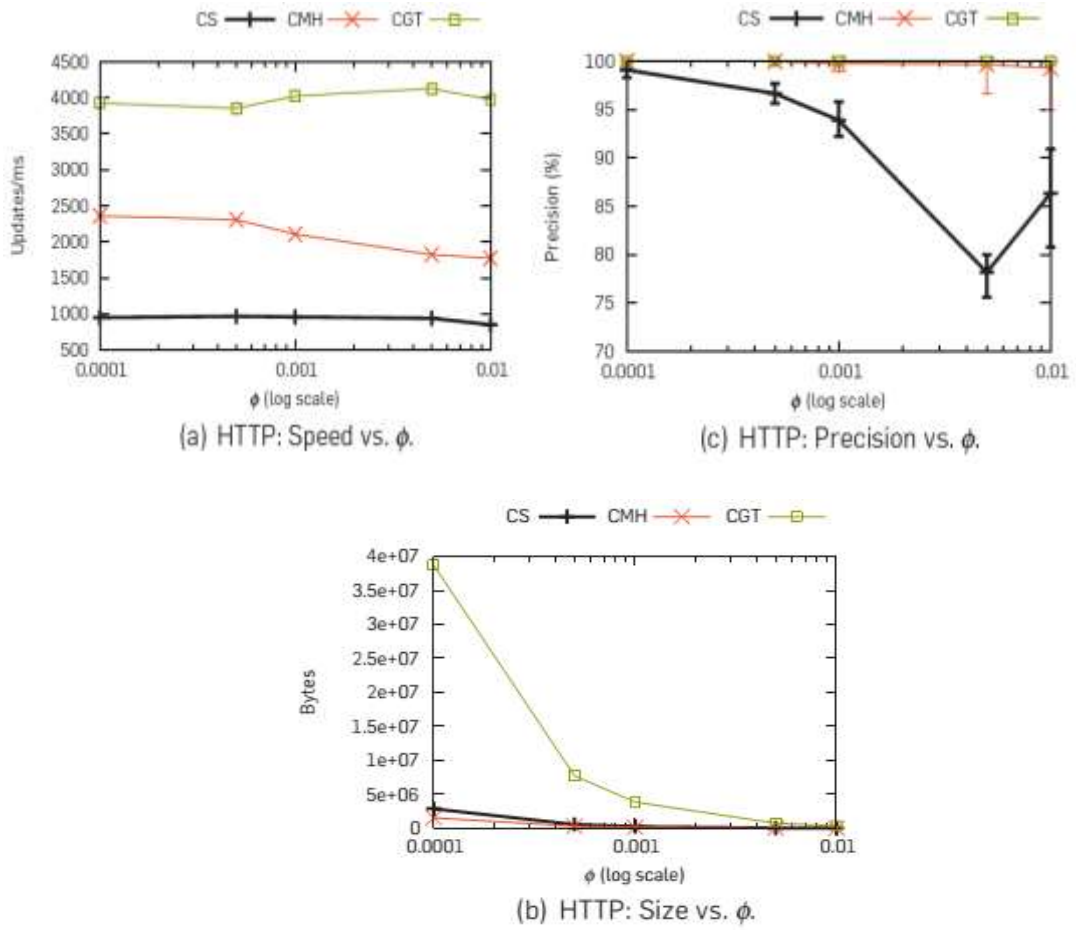
Nhóm nghiên cứu của Cormode và Muthukrishnan sử dụng phương pháp thử nhóm và ứng dụng trong truy xuất cơ sở dữ liệu [10], [27], [36]. Nhóm tác giả cải tiến phương pháp *Count-Min* và lý thuyết thử nhóm để phát hiện các phần tử tần suất cao trong dòng dữ liệu.

Sơ lược về phương pháp được mô tả như sau: giả sử có một dòng dữ liệu *m* phần tử, trong đó *m* rất lớn, thiết kế tổng cộng *t* nhóm thử. Mỗi một nhóm thử thứ *i* sẽ có một bộ đếm c_i tương ứng. Kết quả của nhóm thử có chứa phần tử tần suất cao khi và chỉ khi $c_i > m/(d+1)$, trong đó *d* là một hằng số cho trước.

Nhóm nghiên cứu của Cormode đã mô hình hóa được bài toán phát hiện các phần tử tần suất cao trong dòng dữ liệu và cho thấy được một số ưu điểm vượt trội của phương pháp thử nhóm như phù hợp đối với loại dữ liệu động, có độ chính xác cao và thời gian chạy nhanh hơn các phương pháp khác (*Count-Sketch, Count-Min*). Tuy nhiên, giải pháp này còn có một số hạn chế mà nhóm nghiên cứu của Cormode chưa giải quyết được. Đó là phương pháp sinh ma trận phân cách theo phương pháp xác suất, dẫn đến vấn đề ma trận sinh ra có thể không phải là ma trận d-phân-cách hoặc chi phí thời gian do phải vét cạn của thuật toán tham lam. Đây cũng là hạn chế chung của nhiều nhóm nghiên cứu khác như trong [10], [37], [38]. Một số kết quả

thực nghiệm so sánh giữa các thuật toán "sketch" và phương pháp thử nhóm được trình bày ở hình 1.12.

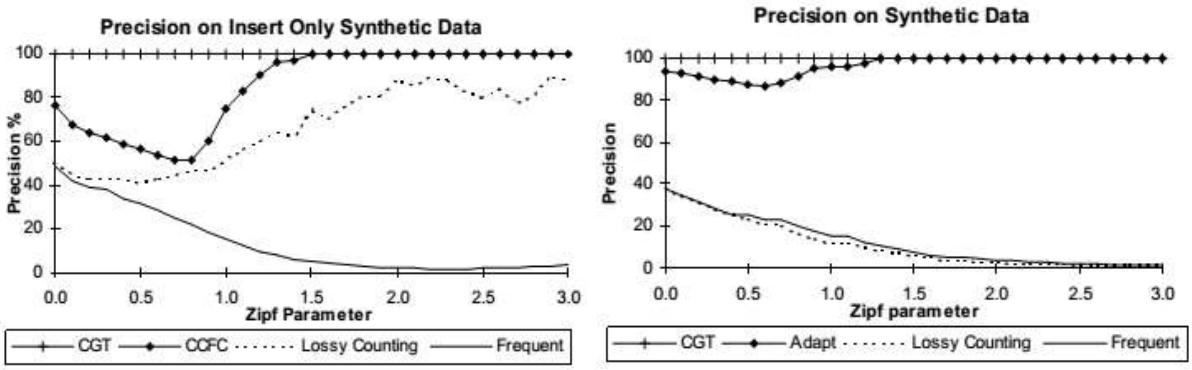
Qua biểu đồ kết quả thực nghiệm cho thấy thuật toán *Count-Sketch* có tốc độ cập nhật chậm hơn các thuật toán khác trong nhóm các thuật toán *sketch*. Thuật toán thử nhóm bất ứng biến (CGT) sử dụng không gian lưu trữ nhiều hơn nhưng thực hiện nhanh và độ chính xác cao.



CS: CountSketch; **CMH**: CountMin Sketch; **CGT**: Combinatorial Group Testing

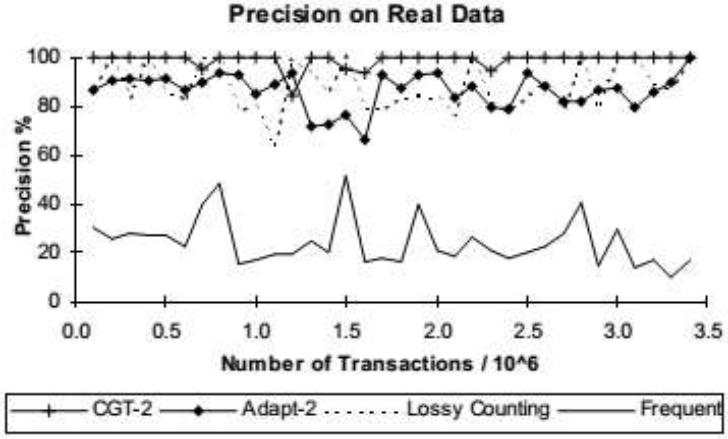
Hình 1.12. *So sánh các thuật toán loại "sketch"* [10]

Một số kết quả thực nghiệm nữa được trình bày trong công trình nghiên cứu của Cormode và Muthukrishnan [36] so sánh độ chính giữa phương pháp thử nhóm và các thuật toán của phương pháp *"counter-based"* được thể hiện trong hình 1.13.



Hình 1.13. *Đồ thị so sánh độ chính xác các thuật toán* [36]

Phần thực nghiệm với dữ liệu thời gian thực so sánh độ chính xác của phương pháp thử nhóm bất ứng biến, phương pháp thử nhóm ứng biến và hai thuật toán của loại *"counter-based"* là *LossyCounting* và *Frequent* được trình bày trong hình 1.14 cho thấy phương pháp thử nhóm bất ứng biến có độ chính xác cao hơn các phương pháp khác.



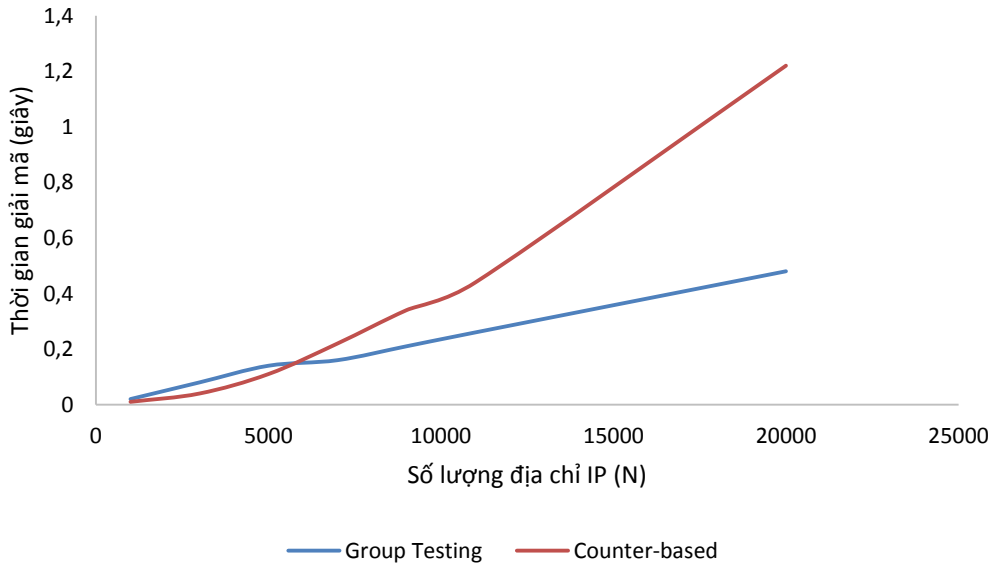
CGT: Combinatorial Group Testing, **Adapt:** Adaptive Group Testing

Hình 1.14. *Đồ thị so sánh độ chính xác các thuật toán trên dữ liệu thật* [36]

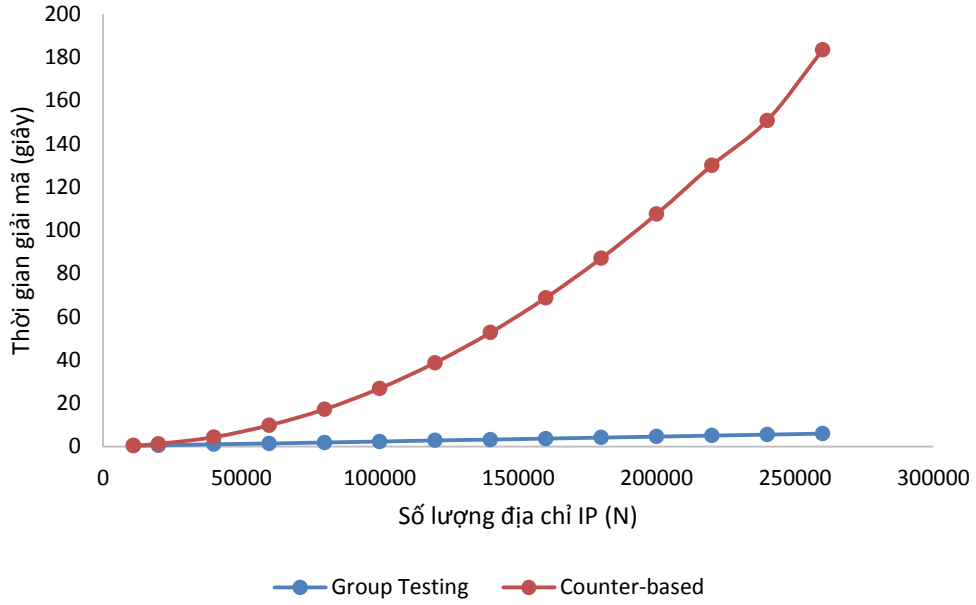
Luận án cũng đã tiến hành cài đặt thực nghiệm so sánh giữa phương pháp "counter-based" và phương pháp thử nhóm bất ứng biến, tính thời gian giải mã với số lượng địa chỉ IP phân biệt từ 3.000 đến 260.000, số lượng gói tin IP được phát sinh ngẫu nhiên trên server (IBM Xeon E 2.5 Ghz, RAM 4GB). Kết quả được trình bày trong bảng 1.3.

Từ kết quả thực nghiệm cho thấy rằng phương pháp *"counter-based"* cho kết quả tốt hơn phương pháp thử nhóm bất ứng biến trong trường hợp số lượng phần tử nhỏ. Tuy nhiên với số lượng phần tử lớn, phương pháp thử nhóm bất ứng biến cho kết quả tốt hơn. Kết quả thực nghiệm thể hiện trong bảng 1.3, hình 1.15 và hình 1.16.

Qua các phân tích trên cho thấy phương pháp thử nhóm bất ứng biến có nhiều ưu điểm trong bài toán tìm phần tử tần suất cao trên mạng như thực hiện đơn giản, tốc độ nhanh và độ chính xác cao. Phương pháp này có khả năng ứng dụng hiệu quả cho bài toán phát hiện các Hot-IP trực tuyến trên mạng, đặc biệt ở mạng trung gian ở phía nhà cung cấp dịch vụ với số lượng người dùng và tần suất sử dụng rất lớn. Trong đó có thể coi các địa chỉ IP trong các gói tin IP là các phần tử cần xem xét.



Hình 1.15. *Biểu đồ thời gian giải mã của "Group Testing" và "counter-based"*



Hình 1.16. *Biểu đồ thời gian giải mã của "Group Testing" và "counter-based" với số lượng đối tượng lớn*

Bảng 1.3. *Thời gian giải mã của phương pháp thử nhóm và "counter-based"*

N	**Group Testing** (giây)	**Couter-Based** (giây)	**N**	**Group Testing** (giây)	**Couter-Based** (giây)
3.000	0,08	0,04	100.000	2,28	26,89
5.000	0,14	0,11	120.000	2,79	38,73
7.000	0,16	0,22	140.000	3,19	52,67
9.000	0,21	0,34	160.000	3,65	68,78
11.000	0,26	0,44	180.000	4,10	87,05
20.000	0,48	1,22	200.000	4,56	107,49
40.000	1,01	4,31	220.000	5,01	130,05
60.000	1,37	9,79	240.000	5,48	150,78
80.000	1,84	17,17	260.000	5,93	183,38

1.5. GIẢI PHÁP PHÁT HIỆN HOT-IP

Xuất phát từ hai bài toán ứng dụng thực tế là bài toán tấn công từ chối dịch vụ và bài toán phát tán sâu trên Internet cho thấy đặc trưng quan trọng của chúng là số lượng gói tin lưu thông trên mạng rất lớn trong khoảng thời gian rất ngắn. Mỗi gói tin IP lưu thông trên mạng đều chứa thông tin về địa chỉ IP bên máy gửi và IP bên máy nhận. Do đó, bài toán phát hiện các đối tượng trên mạng xuất hiện với tần suất cao; liên quan đến các đối tượng có khả năng là nguồn phát hay mục tiêu trong tấn công từ chối dịch vụ, có khả năng là đối tượng đang tiến hành phát tán sâu Internet loại quét không gian địa chỉ IP để tìm kiếm các mục tiêu lây nhiễm; có thể tổng quát thành bài toán phát hiện các Hot-IP trên mạng.

Phát hiện các Hot-IP là một trường hợp trong bài toán tìm các phần tử tần suất cao trong dòng dữ liệu. Trên cơ sở phân tích các nghiên cứu liên quan và các thuật toán phát hiện phần tử tần suất cao trên dòng dữ liệu cho thấy rằng phương pháp thử nhóm bất ứng biến có nhiều lợi thế để áp dụng vào việc phát hiện các Hot-IP trực tuyến trên mạng.

Có thể mô hình hóa bài toán phát hiện các Hot-IP trên mạng như sau: cho không gian rất lớn các địa chỉ IP. Mỗi gói tin trên mạng thông qua địa chỉ IP để xác định thông tin người gửi và người nhận trên mạng, các thiết bị định tuyến là thành phần trung gian chuyển tiếp các gói tin đến đích dựa vào thông tin địa chỉ này và bảng định tuyến. Các nguy cơ gây hại xuất phát từ một hoặc một số lượng đối tượng nào đó (Hot-IP) rất nhỏ so với số lượng các thiết bị hoạt động bình thường trên mạng cần được xác định để có giải pháp ứng phó kịp thời. Mục tiêu của luận án là đưa ra giải pháp phát hiện các Hot-IP trực tuyến với dòng dữ liệu lớn. Một số vấn đề cần xem xét là: không gian lưu trữ, thời gian tính toán, phương pháp bố trí bộ phát hiện Hot-IP phân tán cho các hệ thống mạng đa vùng, lựa chọn các tham số cho giải pháp theo vị trí triển khai và khả năng của hệ thống.

❖ **Một số nghiên cứu về thuật toán giải mã trong phương pháp thử nhóm:**

Trong các nghiên cứu về giải pháp thử nhóm bất ứng biến, các nghiên cứu liên quan về thuật toán giải mã tìm ra các phần tử tần suất cao (Hot-IP) cho đến nay có 2 thuật toán được đề cập chủ yếu là: *thuật toán giải mã đơn giản* và *thuật toán giải mã danh sách*.

Thuật toán giải mã đơn giản (naïve algorithm): dựa vào kết quả của phép thử nhóm và ma trận d-phân-cách để xác định các Hot-IP. Xét các kết quả của các nhóm thử là *"âm tính"* nghĩa là các nhóm thử không chứa Hot-IP thì loại các IP thuộc nhóm này. Sau khi xem xét hết các kết quả *"âm tính"* và loại các IP tương ứng trong các nhóm này thì các địa chỉ IP còn lại là các Hot-IP cần tìm. Thuật toán này đơn giản, tuy nhiên thời gian chạy là O(*Nt*), với *t* là số dòng của ma trận d-phân-cách (số lượng nhóm thử) và *N* là số cột của ma trận d-phân-cách (số lượng địa chỉ IP phân biệt) và d là số lượng Hot-IP tối đa mà giải pháp có thể phát hiện được.

Thuật toán giải mã danh sách: nghiên cứu của nhóm tác giả Indyk-Ngo-Rudra [39][40] cải tiến ý tưởng của Kautz và Singleton trong việc xây dựng ma trận d-phân-cách với số hàng giảm đi và ma trận cho phép giải mã nhanh. Ý tưởng chính của Indyk-Ngo-Rudra là xây dựng các ma trận *(d,d)-phân-cách-danh-sách* M_i dùng làm mã trong với kích thước $n_2 \times q$, nếu ma trận ngoài *M* là một ma trận d-phân-cách thì ta có thể giải mã *M* trong thời gian $poly(d) \cdot t \log^2 t + O(t^2)$ và số hàng của ma trận là *t*, với $t = O(d^2 \log N)$. Một ma trận nhị phân M kích thước $t \times N$ được gọi là một ma trận (d,l)-phân-cách-danh-sách nếu thoả tính chất sau đây: lấy một tập *S* có nhiều nhất *d* cột của *M*, và một tập *T* (không giao với *S*) với ít nhất *l* cột của *M* thì tồn tại ít nhất một hàng *i* của M mà trong đó một cột nào đó trong *T* chứa số 1 còn tất cả các cột khác trong *S* chứa số 0. Thời gian giải mã và phương pháp xây dựng ma trận phân cách được tóm tắt trong bảng 1.4.

Kết quả được nhóm nghiên cứu [39][40] chứng minh phương pháp giải mã danh sách cho kết quả giải mã tốt hơn phương pháp giải mã đơn giản. Tuy nhiên, nhóm tác giả này không chỉ ra cách xây dựng ma trận phân cách danh sách một

cách tường minh mà sử dụng phương pháp xác suất để sinh ma trận. Nghiên cứu này chỉ mang tính chất lý thuyết, khó khăn trong triển khai thực tế.

Một số nghiên cứu khác để tối ưu số hàng của ma trận với $t = O(d^2 \log N)$ được trình bày trong [41], [42]. Tuy nhiên, các phương pháp này không có cách giải mã nhanh và xây dựng được ma trận tường minh. Mô hình 2 bước của phương pháp thử nhóm trình bày trong [43] không thích hợp cho xử lý thời gian thực bởi vì các bài toán trên dòng gói tin IP thời gian thực cần phải được tính toán một lần để cho ra kết quả.

Kết quả nghiên cứu trong [41], [42], [43] chỉ có ý nghĩa về mặt lý thuyết vì không chỉ ra cách xây dựng ma trận này một cách hiệu quả, nhóm tác giả này dùng phương pháp sinh ma trận ngẫu nhiên. Điều này dẫn đến việc phải lưu trữ toàn bộ ma trận trong quá trình thực thi chương trình. Chúng ta chỉ có thể áp dụng phương pháp này trong trường hợp đặc biệt với ma trận d-phân-cách thì cũng là *(d,1)*-phân-cách-danh-sách. Khi đó, thuật toán chỉ có thể phát hiện được tối đa một Hot-IP trên mạng. Trong thực tế triển khai, việc phát hiện nhiều Hot-IP cùng lúc sẽ có nhiều ý nghĩa hơn. Để đáp ứng điều này, phương pháp thử nhóm bất ứng biến cần được cải tiến và có thể kết hợp với một số kỹ thuật khác để nâng cao hiệu quả phát hiện Hot-IP. Hai kỹ thuật quan trọng được xem xét kết hợp đó là kỹ thuật xử lý song song được dùng trong việc tính vector kết quả và kiến trúc phân tán giữa các khu vực trong hệ thống mạng đa vùng.

Bảng 1.4. *Xây dựng ma trận d-phân-cách*

	Kautz-Singleton [45]	**Indyk-Ngo-Rudra [39]**
Số nhóm thử	$t = O(d^2 \log^2 N)$	$t = O(d^2 \log N)$
Thời gian giải mã	$O(tN)$	$poly(d) \cdot t \log^2 t + O(t^2)$
Phương pháp xây dựng	Nonrandom	Random

Qua các phân tích trên cho thấy rằng phương pháp giải mã của Kautz-Singleton còn lớn tuy nhiên phương pháp xây dựng ma trận d-phân-cách lại là phương pháp đại số cho phép phát sinh từng cột của ma trận mà không cần phải lưu trữ toàn bộ ma trận khi thực thi chương trình. Phương pháp giải mã danh sách của Indyk-Ngo-Rudra tối ưu hơn về cách giải mã tuy nhiên việc xây dựng ma trận d-phân-cách dựa vào xác suất trên cơ sở của ma trận (d,d)-phân-cách-danh-sách. Do vậy, để áp dụng hiệu quả cần cải tiến phương pháp giải mã của Kautz-Singleton mà vẫn giữ nguyên phương pháp sinh ma trận.

❖ **Một số ứng dụng của phương pháp thử nhóm:**

Ứng dụng đầu tiên của phương pháp thử nhóm là phát hiện các quân nhân bị bệnh giang mai trong chiến tranh thế giới thứ II [46]. Thay vì phải thử máu từng người để phát hiện bệnh thì tiến hành thử từng nhóm. Nếu nhóm nào cho kết quả âm tính thì tất cả những người thuộc nhóm này không mắc bệnh. Nếu kết quả dương tính thì có ít nhất một người trong nhóm này bị bệnh. Phương pháp này làm giảm đáng kể số lượng phép thử và thời gian thực hiện nhanh.

Phương pháp thử nhóm còn được ứng dụng trong nhiều lĩnh vực khác nhau như trong [53]. Ứng dụng phương pháp thử nhóm bất ứng biến để phát hiện nguồn phát tấn công từ chối dịch vụ được đề cập trong một số nghiên cứu [37], [38]. Công trình nghiên cứu của nhóm Khattab và các công sự năm 2008 ứng dụng phương pháp thử nhóm để phát hiện nguồn phát tấn công từ chối dịch vụ là công trình nghiên cứu ứng dụng đầu tiên của lý thuyết thử nhóm vào bài toán phát hiện đối tượng trong tấn công từ chối dịch vụ [38].

Phương pháp “Live Baiting” là giải pháp hiệu quả được đề xuất cho bài toán phát hiện các kẻ tấn công DDoS trong dịch vụ Web dựa vào phương pháp thử nhóm. Phương pháp này có ưu điểm là giảm quá tải trong quá trình xử lý dòng dữ liệu lớn và không yêu cầu sử dụng đến các mô hình bình thường được thiết lập trước hoặc thiết lập các hành vi bất thường.

Nhóm tác giả thử nghiệm trên phần mềm giả lập NS-2 sau khi thu thập dữ liệu từ Web, thời gian phát hiện trong 90 giây. Trong giải pháp này, các yêu cầu xử lý được đưa vào các nhóm ứng dụng khác nhau, sinh ma trận bằng phương pháp xác suất. Tác giả phân loại dịch vụ thành các lớp. Mỗi lớp *i* của một dịch vụ yêu cầu cung cấp, server có khả năng xử lý là c_i. Các tác giả thử nghiệm với 10.000 client. Sau khi phân tích các lớp dịch vụ, cuối cùng suy ra địa chỉ IP của yêu cầu đó là từ IP nào trên mạng.

Hạn chế của "Live baiting" là thuật toán phát hiện với danh sách nghi ngờ chứa tất cả các client, xác suất dương tính giả cao, đặc biệt khi số lượng người dùng lớn. Hạn chế nữa là ngưỡng được thiết lập cố định dựa trên số lượng yêu cầu cho mỗi nhóm thử trong suốt thời gian giám sát mà không xem xét diễn biến lưu lượng thay đổi. Nhóm nghiên cứu của Khattab cải tiến phương pháp "Live baiting" với ngưỡng có thể thay đổi trong quá trình thực hiện để cải tiến mức độ chính xác của thuật toán trong [44].

Một số vấn đề cần cải tiến từ nghiên cứu của nhóm Khattab là: (1) có thể chỉ cần tập trung vào việc trích thông tin IP ở tầng mạng trên các thiết bị định tuyến mà không cần phải xử lý ở mức ứng dụng sẽ tiết kiệm thời gian hơn, (2) ma trận Khattab sử dụng được sinh bằng phương pháp xác suất, dẫn đến khả năng ma trận sinh ra không phải là d-phân-cách. Từ đó, kết quả giải mã xảy ra tình trạng mức độ chính xác không cao.

Nhóm nghiên cứu của Ying Xuan & Thai năm 2010 cũng nghiên cứu về thử nhóm ứng dụng để phát hiện tấn công từ chối dịch vụ, cài đặt trên các máy chủ bên trong [37]. Trong đó, các máy chủ ảo được dùng như là các nhóm thử, các yêu cầu sẽ được phân bố vào các máy chủ ảo này.

Một số vấn đề cần cải tiến của nhóm Ying Xuan & Thai là: số lượng máy chủ ảo tương ứng với số nhóm thử mà nhóm nghiên cứu sử dụng làm cho số lượng nhóm thử nhỏ (vì khả năng sử dụng các máy chủ ảo là có giới hạn, thường là nhỏ) so với số đối tượng trên mạng tương ứng với số cột của ma trận. Do đó dẫn đến

mức độ chính xác của phương pháp thử nhóm không cao. Vì trong phương pháp thử nhóm số lượng phép thử $t=O(d^2\log^2 N)$. Trong đó, *t* là số lượng phép thử, *d* là số lượng tấn công cũng là d trong ma trận d-phân-cách, *N* là số lượng đối tượng trên mạng.

Các nghiên cứu hiện tại về phương pháp thử nhóm bất ứng biến và các ứng dụng của nó còn có một số hạn chế cần được tiếp tục nghiên cứu cải tiến như sau:

(1) Vấn đề xây dựng ma trận d-phân-cách: cần có cách xây dựng tường minh, chính xác sẽ đảm bảo được việc giải mã chính xác. Số lượng địa chỉ IP trên mạng rất lớn (khoảng 2^{32} địa chỉ IPv4, 2^{128} đối với địa chỉ IPv6), đặc biệt trên mạng ở phía nhà cung cấp dịch vụ. Do đó, việc lưu trữ số lượng phần tử sẽ chiếm nhiều không gian bộ nhớ. Nếu có phương pháp phát sinh từng cột của ma trận thì sẽ giải quyết được vấn đề này, phương pháp này có thể cho phép cứng hóa bước sinh ma trận.

Có 2 phương pháp xây dựng ma trận d-phân-cách: phương pháp thứ nhất là phương pháp xác suất với $t = O(d^2 \log N)$ [39], phương pháp này không thể sinh ra từng cột của ma trận mà chúng ta muốn xử lý. Thứ hai là phương pháp xây dựng tường minh sử dụng phép nối mã của Kautz và Singleton với $t = O(d^2 \log^2 N)$ [45]. Đây là giải pháp cân bằng giữa phương pháp sinh ma trận dạng *random* và *nonrandom* về số lượng nhóm thử.

(2) Kết hợp với một số kỹ thuật để tăng hiệu quả giải pháp phát hiện Hot-IP như lựa chọn kích thước ma trận phù hợp với vị trí triển khai, xử lý song song, kiến trúc phân tán trong các hệ thống mạng tổ chức đa vùng.

(3) Cải tiến phương pháp thử nhóm bất ứng biến để nâng cao khả năng tính toán và phù hợp với bài toán phát hiện các Hot-IP thời gian thực, đảm bảo hệ thống hoạt động ổn định, thông suốt.

1.6. KẾT LUẬN CHƯƠNG 1

Trong chương này, luận án trình bày tổng quan về các thuật toán tìm phần tử tần suất cao trong dòng dữ liệu, các nghiên cứu liên quan đến bài toán phát hiện và xác định các đối tượng trong tấn công từ chối dịch vụ (DoS, DDoS), các nghiên cứu về phát hiện đối tượng phát tán sâu Internet loại *"scanning worm"*. Trong vấn đề tìm phần tử tần suất cao, luận án trình bày về Hot-IP, một số đặc điểm của Hot-IP trong dòng gói tin IP, giải pháp phát hiện và các vấn đề nghiên cứu đặt ra cho bài toán này khi áp dụng vào dòng dữ liệu thời gian thực. Một số phân tích liên quan đến việc lựa chọn phương pháp cho bài toán phát hiện các phần tử tần suất cao và những thách thức trong các ứng dụng thời gian thực trên dòng dữ liệu như chi phí tính toán và không gian lưu trữ vốn hạn chế trên các thiết bị mạng.

Hạn chế trong các nghiên cứu đã khảo sát là ở bước phát hiện tấn công, các giải pháp chỉ tập trung vào phát hiện có tấn công hay không trong dòng dữ liệu mà không chỉ ra các đối tượng gây ra tấn công bằng các kỹ thuật phân tích thống kê, kỹ thuật khai phá dữ liệu, phương pháp học máy. Sử dụng phương pháp "dò ngược" để phát hiện các đối tượng tấn công rất khó áp dụng trong môi trường Internet và phương pháp này thường thực hiện ở giai đoạn hậu tấn công.

Bài toán phát hiện các Hot-IP trực tuyến là bài toán có nhiều ứng dụng quan trọng trên mạng như phát hiện các thiết bị có khả năng hoạt động bất thường, phát hiện các đối tượng có khả năng là mục tiêu trong tấn công từ chối dịch vụ, phát hiện các đối tượng có khả năng là nguồn phát động tấn công từ chối dịch vụ hay phát hiện nguồn phát tán sâu mạng. Việc triển khai ứng dụng giải pháp này trên hệ thống mạng lớn có rất nhiều người truy cập hay mạng ở phía nhà cung cấp dịch vụ có ý nghĩa quan trọng nhằm phát hiện sớm các đối tượng có khả năng gây nguy hại trên mạng để từ đó giúp người quản trị có những giải pháp ứng phó kịp thời, đảm bảo hệ thống hoạt động ổn định, thông suốt.

Phương pháp thử nhóm bất ứng biến có nhiều ưu điểm để nghiên cứu triển khai với dòng dữ liệu lớn thời gian thực trên mạng như thời gian thực hiện nhanh,

độ chính xác cao và đơn giản. Tuy nhiên, hạn chế lớn nhất của phương pháp này là còn chiếm nhiều không gian lưu trữ ma trận.

Bài toán này mở ra nhiều hướng nghiên cứu để cải tiến giảm không gian lưu trữ bằng cách xây dựng tường minh ma trận phân cách sử dụng phương pháp nối mã. Khi đó, chương trình không cần phải lưu trữ ma trận mà vẫn xác định được giá trị trong các phần tử của nó. Bên cạnh đó, giải pháp có thể được cải tiến giảm thời gian tính toán để ứng dụng trong phát hiện các Hot-IP trực tuyến và có thể kết hợp một số giải pháp khác như kỹ thuật xử lý song song, kiến trúc phân tán, xem xét khả năng của hệ thống tại vị trí triển khai để nâng cao hiệu quả phát hiện nhanh và cảnh báo sớm các Hot-IP.

CHƯƠNG 2. PHÁT HIỆN CÁC HOT-IP SỬ DỤNG THỬ NHÓM BẤT ỨNG BIẾN

2.1. GIỚI THIỆU VỀ THỬ NHÓM

Phương pháp thử nhóm xuất hiện đầu tiên vào năm 1943 được Robert Dorfman đề xuất [46]. Trong chiến tranh thế giới lần thứ II, Dorfman thiết kế một thủ tục thử máu cho các quân nhân của Mỹ để chỉ ra những người nào bị bệnh giang mai. Ông thực hiện như sau: bỏ nhiều mẫu máu vào các nhóm, mỗi mẫu máu được trích ra để bỏ vào nhiều nhóm. Mỗi nhóm có thể chứa một hoặc nhiều mẫu máu và thử cùng một lúc. Giả sử bỏ qua vấn đề dương tính giả và cho rằng các phép thử máu không bị lỗi. Nếu phép thử là *âm tính* thì tất cả các mẫu máu trong nhóm đó là *âm tính*. Nếu phép thử là *dương tính* thì có ít nhất có một mẫu máu trong nhóm đó là *dương tính*. Vấn đề đặt ra là cho trước *N* mẫu máu, thiết kế các nhóm thử càng ít càng tốt để chỉ ra các mẫu máu *dương tính*. Ý tưởng *thử nhóm* làm giảm đáng kể tổng số các phép thử. Mục tiêu của phương pháp thử nhóm là xác định một tập con *d* các phần tử *"dương tính"* từ một tập rất lớn *N* các đối tượng với số lần thử càng ít càng tốt.

Lý thuyết thử nhóm ra đời từ đó và có nhiều ứng dụng quan trọng trong nhiều lĩnh vực khác nhau [47], [53]: trong sinh học tính toán [48], mạng máy tính [49], xử lý tín hiệu [50], xác minh chữ ký điện tử theo nhóm [51], dữ liệu trực tuyến [27], kiểm tra tính toàn vẹn dữ liệu [52].

Phương pháp thử nhóm được chia thành 2 loại là *thử nhóm ứng biến* (Adaptive Group Testing) và *thử nhóm bất ứng biến* (Non-Adaptive Group Testing – NAGT) [53]. Trong thử nhóm ứng biến, phép thử sau được thiết kế dựa vào kết quả của phép thử trước đó, thuật toán thử nhóm ứng biến có bản chất tuần tự. Trong thử nhóm bất ứng biến, tất cả các phép thử phải được xác định trước mà không phụ thuộc vào bất kỳ phép thử nào. Đây cũng là yếu tố quan trọng có thể thực hiện việc chạy song song các phép thử trên nhiều bộ xử lý cùng một lúc nhằm giảm thời gian

tính toán giúp phát hiện nhanh các Hot-IP trên mạng. Trong một số ứng dụng cho các bài toán trên dòng dữ liệu yêu cầu phải sử dụng phương pháp thử nhóm bất ứng biến vì dữ liệu trên dòng dữ liệu đi qua thuật toán một lần và cho ra kết quả ngay. Do đó, luận án chỉ tập trung nghiên cứu về phương pháp thử nhóm bất ứng biến để áp dụng vào bài toán phát hiện các Hot-IP trực tuyến trên mạng.

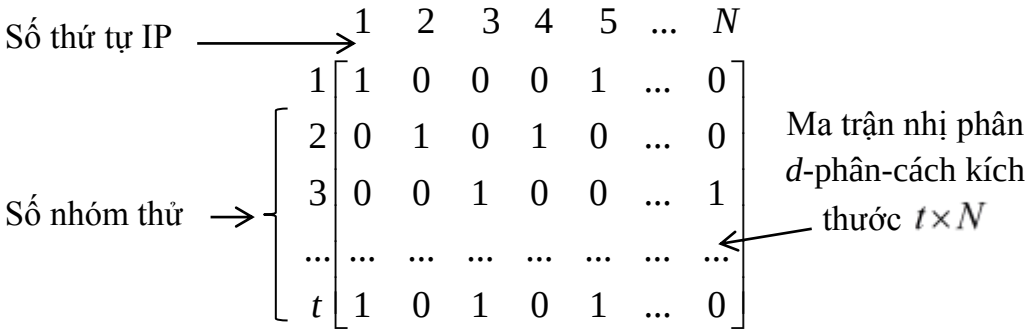
Phương pháp thử nhóm bất ứng biến có nhiều ưu điểm như tính đơn giản, thực hiện nhanh, độ chính xác cao, khả năng ứng dụng hiệu quả trong dòng dữ liệu lớn như đã phân tích ở chương 1. Luận án áp dụng phương pháp thử nhóm bất ứng biến vào bài toán phát hiện các Hot-IP trực tuyến trên dòng gói tin IP, đề xuất thuật toán cải tiến, kết hợp với một số kỹ thuật như xử lý song song và kiến trúc phân tán để nâng cao hiệu quả của giải pháp. Giải pháp phát hiện các Hot-IP trên mạng có thể ứng dụng vào một số bài toán an ninh mạng như phát hiện các đối tượng có khả năng là nguồn phát hay mục tiêu trong các tấn công từ chối dịch vụ, phát hiện các đối tượng có khả năng là nguồn phát tán sâu Internet, phát hiện các đối tượng có khả năng đang hoạt động bất thường trong hệ thống, có thể triển khai giải pháp ở các mạng trung gian như mạng của các nhà cung cấp dịch vụ hoặc các hệ thống mạng cung cấp dịch vụ trên Internet.

2.2. THỬ NHÓM BẤT ỨNG BIẾN

Trong thử nhóm bất ứng biến, các nhóm thử phải được thiết kế trước, thử tất cả các nhóm cùng một lúc, rồi từ đó chỉ ra các Hot-IP [39]. Phương pháp này thích hợp cho các bài toán trên dòng dữ liệu thời gian thực, khi đó với dữ liệu đầu vào thuật toán chỉ cần thực hiện việc tính toán một lần để cho ra kết quả.

Mô hình hóa bài toán phát hiện các Hot-IP trên dòng gói tin IP về bài toán thử nhóm bất ứng biến như sau: cho dòng gói tin IP, trong đó có N địa chỉ IP phân biệt. Giả sử có tối đa d phần tử là Hot-IP, thiết kế t nhóm thử cho N địa chỉ IP này. Xây dựng một ma trận nhị phân M_{txN}, trong đó các cột của ma trận đại diện cho các địa chỉ IP phân biệt và các hàng của ma trận đại diện cho các nhóm thử. Các phần tử

của ma trận m_{ij} có giá trị như sau: $m_{ij}=1$ nghĩa là IP thứ *j* thuộc về nhóm thử thứ *i* và ngược lại $m_{ij}=0$ nếu IP thứ *j* không thuộc về nhóm thử thứ *i*.



Hình 2.1. *Ma trận nhị phân d-phân-cách*

Nếu *M* là ma trận d-phân-cách thì có thể chỉ ra rằng có nhiều nhất *d* phần tử là Hot-IP,với $d \ll N, t \ll N$, nghĩa là tổng không gian sử dụng để lưu trữ trong phương pháp thử nhóm nhỏ hơn rất nhiều so với phương pháp dùng mỗi bộ đếm cho mỗi IP [53]. Hình 2.1 mô tả ma trận nhị phân d-phân-cách kích thước $t \times N$. Để chỉ ra các Hot-IP trong dòng gói tin IP, từ ma trận d-phân-cách và vector kết quả của các nhóm thử, thuật toán giải mã sẽ chỉ ra những địa chỉ IP nào là Hot-IP mà không cần bất kỳ một cấu trúc dữ liệu nào khác [39].

Gọi thuật toán xác định các Hot-IP là thuật toán giải mã và thời gian chạy thuật toán tìm ra các Hot-IP là thời gian giải mã. Thuật toán giải mã phổ biến được sử dụng là thuật toán giải mã đơn giản. Có thể tóm tắt thuật toán giải mã này như sau: cho một vector kết quả $r = (r_i) \in \{0,1\}^t$, nếu IP thứ j nằm trong nhóm thử *"âm tính"* thứ *i* (nghĩa là $r_i = 0$ và $m_{ij} = 1$ thì *j* không phải là Hot-IP). Sau khi loại bỏ hết các IP không phải là Hot-IP theo cách này thì kết quả thu được còn lại là các Hot-IP.

Một số yêu cầu quan trọng đối với các thuật toán sử dụng phương pháp thử nhóm bất ứng biến cho các ứng dụng trong dòng dữ liệu: *thiết kế số lượng nhóm thử nhỏ, thời gian xác định các Hot-IP nhanh, xây dựng ma trận d-phân-cách tường minh và giảm không gian lưu trữ* [43], [53].

❖ *Về số lượng nhóm thử*

Số lượng nhóm thử thể hiện số phép kiểm tra nhóm phải thực hiện, từ đó chỉ ra kết quả là những IP nào là Hot-IP trong tổng số N địa chỉ IP. Theo lý thuyết thử nhóm số lượng nhóm thử được thiết kế càng nhỏ càng tốt.

Năm 1964, Kautz và Singleton đề ra cách thiết kế các nhóm chỉ cần số phép thử là $O(d^2\log^2 N)$ [45]. Trong đó *d* là chặn trên của số mẫu *"dương tính"* và *N* là tổng số mẫu (*d* rất nhỏ so với *N*). Họ nghiên cứu các mã chồng (superimposed codes) và chứng minh được rằng tồn tại các phép thiết kế chỉ cần số nhóm thử là $O(d^2\log N)$, nhưng không chỉ ra được cách xây dựng. Ý tưởng của họ là dùng các mã phân ly khoảng các tối đa (MDS) để xây dựng các nhóm thử. Ý tưởng này là trường hợp đặc biệt của phép nối mã khá phổ biến trong lý thuyết mã hóa. Mã MDS phổ biến nhất là mã Reed-Solomon.

❖ *Về xây dựng ma trận d-phân-cách tường minh*

Xây dựng ma trận d-phân-cách là một cơ sở quan trọng trong phương pháp thử nhóm bất ứng biến. Cách xây dựng ma trận này mô tả ba tham số quan trọng: số hàng của ma trận thể hiện số nhóm thử, số cột của ma trận thể hiện số lượng địa chỉ IP phân biệt có thể hỗ trợ và tham số *d* trong ma trận nhị phân d-phân-cách thể hiện số lượng Hot-IP tối đa có thể tìm ra được bởi phương pháp thử nhóm bất ứng biến.

Có 2 phương pháp xây dựng ma trận d-phân-cách được đề xuất: phương pháp sinh ngẫu nhiên và phương pháp sinh tường minh sử dụng phép nối mã [39]. Phương pháp sinh ngẫu nhiên có tính xác suất, nghĩa là có thể phát sinh ma trận không phải là ma trận d-phân-cách. Hơn nữa, ma trận phải được phát sinh toàn bộ, nghĩa là phải lưu toàn bộ ma trận trong bộ nhớ khi thực thi chương trình, tốn không gian lưu trữ. Phương pháp sinh ma trận tường minh đảm bảo ma trận sinh ra chính xác là ma trận d-phân-cách và có thể phát sinh theo từng cột của ma trận dựa vào phép nối mã. Do đó, cách sinh ma trận dạng này đảm bảo tính chính xác của ma trận d-phân-cách để phương pháp giải mã có tính chính xác cao và có thể ứng dụng triển khai giải pháp trên các thiết bị có tài nguyên hạn chế, khi đó các cột của ma

trận được phát sinh và tính toán mà không cần phải lưu toàn bộ ma trận vào bộ nhớ khi thực thi chương trình.

❖ *Về thời gian xác định các Hot-IP*

Gọi thời gian giải mã là thời gian chạy thuật toán để tìm ra các Hot-IP trong dòng dữ liệu. Thuật toán giải mã trong phương pháp thử nhóm bất ứng biến sử dụng ma trận d-phân-cách và vector kết quả của các nhóm thử để xác định các Hot-IP mà không cần một cấu trúc dữ liệu nào khác. Trong các bài toán trên dòng dữ liệu thời gian thực thì yếu tố thời gian là rất quan trọng để triển khai áp dụng vào thực tế.

Hai thuật toán quan trọng được sử dụng là *thuật toán giải mã đơn giản* và *thuật toán giải mã danh sách*. Thuật toán giải mã đơn giản có thời gian giải mã là $O(tN)$ và thuật toán giải mã danh sách của nhóm Indyk-Ngo-Rudra là $poly(d)\cdot t\log^2 t + O(t^2)$ [39]. Như vậy, thuật toán giải mã danh sách tối ưu hơn về thời gian giải mã. Tuy nhiên, phương pháp giải mã danh sách khó khăn trong xây dựng ma trận d-phân-cách vì để phát sinh được theo cách này yêu cầu phải xây dựng các mã trong là các ma trận phân cách danh sách khó thực hiện [39].

❖ *Về không gian lưu trữ*

Không gian lưu trữ được nhắc đến ở đây chính là không gian lưu trữ ma trận d-phân-cách. Phạm vi áp dụng của bài toán phát hiện các Hot-IP là trên các mạng có số lượng đối tượng và tần suất truy cập rất lớn, như mạng trung gian của các nhà cung cấp dịch vụ, do vậy với số lượng IP lớn trong dòng gói tin IP đồng nghĩa với việc kích thước ma trận sẽ lớn. Hai vấn đề cần được xem xét giải quyết để phát hiện các Hot-IP trực tuyến là phát sinh từng cột của ma trận để tính toán nhằm giảm không gian lưu trữ và giới hạn số lượng IP của ma trận.

Để giảm không gian lưu trữ cho ma trận, phương pháp nối mã được sử dụng để phát sinh các cột của ma trận. Từng cột của ma trận được sinh ra để sử dụng cho các tính toán trong chương trình mà không cần phải lưu trữ toàn bộ ma trận d-phân-cách có kích thước rất lớn này.

Đối với vấn đề thứ hai là giới hạn số lượng IP (tương ứng với số cột trong ma trận phân cách), có thể xem xét ở vị trí triển khai về số lượng khách hàng, khả năng của thiết bị triển khai giải pháp và chu kỳ thực hiện thuật toán để lựa chọn phù hợp. Vấn đề này sẽ được luận án trình bày chi tiết ở phần sau.

2.3. MA TRẬN D-PHÂN-CÁCH

Định nghĩa 2. *Ma trận nhị phân* $M_{t\times N}$ *được gọi là d-phân-cách khi và chỉ khi hội của d cột bất kỳ không chứa bất kỳ một cột nào khác. Với d+1 cột* C_0, C_1,...,C_d *bất kỳ của M, ta có* $C_0 \subsetneq C_1 \cup ... \cup C_d$.

Một dòng IP $a_1, a_2, ..., a_m$ được thu thập, với mỗi gói tin a_i có địa chỉ IP nguồn là s_i. Giả sử rằng có tổng cộng *N* địa chỉ IP phân biệt được ánh xạ thành các cột của ma trận, *t* nhóm thử được thiết lập là các dòng của ma trận, các phần tử trong ma trận được xác định như sau:

$$m_{ij} = \begin{cases} 1 & nếu\ IP_j\ thuộc\ nhóm\ thử\ i \\ 0 & ngược\ lại \end{cases}$$

Ví dụ sau là ma trận 2-phân-cách. Thực vậy, ta có thể kiểm tra hội của 2 cột bất kỳ không chứa bất kỳ cột khác.

$$M_{9\times 7} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}, d = 2, N = 7, t = 9$$

❖ **Xây dựng ma trận d-phân cách:**

Có 2 phương pháp xây dựng ma trận d-phân-cách phổ biến là phương pháp xác suất và phương pháp nối mã. Xây dựng ma trận theo phương pháp xác suất

không đảm bảo được độ chính xác của ma trận phân cách và không tối ưu về không gian lưu trữ khi thực thi chương trình do phải lưu toàn bộ ma trận khi thực thi chương trình. Do đó, luận án chỉ đề cập đến phương pháp xây dựng ma trận d-phân-cách tường minh bằng phép nối mã. Cách xây dựng này cho phép phát sinh ma trận d-phân-cách chính xác và có thể phát sinh ma trận theo từng cột. Điều này có thể áp dụng trên các thiết bị triển khai có tài nguyên hạn chế. Bảng 2.1 mô tả các phương pháp xây dựng ma trận phân cách.

Bảng 2.1. *Các phương pháp xây dựng ma trận d-phân-cách*

Phương pháp	**Nhóm nghiên cứu**	**Đặc điểm của ma trận**
Xác xuất	Indyk-Ngo-Rudra [39] Parat & Rothschild [42]	Phát sinh toàn bộ ma trận
Nối mã	Kautz-Singleton [45]	Phát sinh từng cột của ma trận

Nếu chỉ quan tâm đến số hàng của ma trận nghĩa là tổng số phép thử thì theo D'yachkov và Rykov [55] chứng minh được chặn dưới của số phép thử là:

$$t(d,N) = \Omega(d^2 \frac{\log N}{\log d}) \tag{2.1}$$

Trên thực tế thời gian giải mã, nghĩa là thời gian chỉ ra các Hot-IP cũng là một tham số quan trọng, nhất là các ứng dụng trực truyến. Các phương pháp xây dựng ma trận d-phân-cách nêu trên không chỉ ra cách giải mã như thế nào ngoài thuật toán giải mã đơn giản "naïve algorithm".

❖ **Một số khái niệm:**

Cho một số nguyên $l \geq 1$, ký hiệu $[l]$ là tập các phần tử $\{1,..,l\}$. Ký hiệu $\mathbb{F}_q$ là một trường hữu hạn của q phần tử. Ký hiệu M_j là cột thứ j trong ma trận M kích thước $t \times N$. Cho $q \geq 2$ là một số nguyên.

- ***Mã*** *C* là một tập con của $[q]^n$ với các số nguyên dương *q* và *n*. Các phần tử của *C* gọi là các từ mã. Tham số *q* gọi là kích thước mã của C. Khi *q=2* thì C gọi là mã nhị phân. Tham số *n* gọi là độ dài khối mã.
- ***Khoảng cách Hamming*** giữa hai từ mã có chiều dài bằng nhau là số các ký hiệu ở vị trí tương đương có giá trị khác nhau.

$$d(X,Y) = \sum_{i=1}^{n} (X_i \neq Y_i) \tag{2.2}$$

- ***Khoảng cách của mã*** là khoảng cách Hamming nhỏ nhất giữa hai từ mã. Gọi $\Delta(c_1, c_2)$ là khoảng cách giữa hai vector trong Σ^n, thì

$$\text{dist}(C) = \min_{c_1 \neq c_2 \in C} \Delta(c_1, c_2) \tag{2.3}$$

- ***Mã tuyến tính:***

Ở phần trên, mã *C* được định nghĩa là một tập con của Σ^n, khi đó $\Sigma = \mathbb{F}_q$ là một trường hữu hạn, *q* là lũy thừa của một số nguyên tố, *C* là một không gian con tuyến tính của Σ^n. Một mã tuyến tính với độ dài *n*, số chiều *k* trên trường $\mathbb{F}_q$ được ký hiệu là $[n,k]_q$. Nếu khoảng cách Hamming giữa hai từ mã bất kỳ ít nhất là *d* thì mã này được ký hiệu là $[n,k,d]_q$.

- ***Mã Reed Solomon:***

Mã Reed-Solomon (RS) là một bộ mã $[n,k]_q$ với $k \leq n \leq q,$ được định nghĩa như sau: mỗi một thông điệp $m = (m_0,...,m_{k-1}) \in \mathbb{F}_q^k$ có thể xem như một đa thức bậc $k-1$ trên vành $\mathbb{F}_q[X]$:

$$P_m(X) = m_0 + m_1 X + \cdots + m_{k-1} X^{k-1} \tag{2.4}$$

Mã RS là một ánh xạ $RS : \mathbb{F}_q^k \to \mathbb{F}_q^n$ được xác định như sau: cố định *n* phần tử khác nhau $\alpha_1,...,\alpha_n \in \mathbb{F},$

$$RS(m) = (P_m(\alpha_1), ..., P_m(\alpha_n)) \quad (2.5)$$

Một đa thức bậc $k-1$có nhiều nhất $k-1$ nghiệm trên trường bất kỳ. Hai đa thức khác nhau chỉ có thể trùng giá trị ở nhiều nhất là $k-1$điểm α_i. Mã RS có khoảng cách ít nhất là $d = n-k+1.$Mã RS là một mã $[n,k,n-k+1]_q$. Mã RS đạt đến chặn Singleton và vì thế nó còn được gọi là *mã phân ly khoảng cách tối đa* (mã MDS). Mã RS có rất nhiều ứng dụng trên thực tế như trong CD, DVD, truyền thông vệ tinh.

❖ **Phương pháp nối mã (phép nối mã):**

Phương pháp nối mã được Forney đề xuất vào năm 1965 [16]. Gọi C_{out} là một mã $(n_1,k_1)_q$, nghĩa là một ánh xạ từ $\mathbb{F}_q^{k_1}$ vào $\mathbb{F}_q^{n_1}$, trong đó $q = 2^{k_2}$, như vậy mỗi ký tự mã của C_{out} là một phần tử của tập $\mathbb{F}_q$ gồm 2^{k_2} phần tử và cũng có thể xem như một phần tử của tập $\mathbb{F}_2^{k_2} = \{0,1\}^{k_2}$. Mỗi từ mã của C_{out} có n_1 vị trí. C_{out} gọi là mã ngoài. Xét n_1 mã $(n_2,k_2)_2$ký hiệu là $C_{in}^1, C_{in}^2, ..., C_{in}^{n_1}$. Nghĩa là, với mọi $i \in [n_1]$ thì C_{in}^i là một ánh xạ $C_{in}^i : \mathbb{F}_2^{k_2} \to \mathbb{F}_2^{n_2}$. Các mã $C_{in}^1, C_{in}^2, ..., C_{in}^{n_1}$ được gọi là các mã trong.

Phép nối mã giữa mã ngoài và các mã trong, ký hiệu là $C_{out} \circ (C_{in}^1, C_{in}^2, ..., C_{in}^{n_1})$, là một mã $(n_1 \times n_2, k_1 \times k_2)_2$ được định nghĩa như sau: cho một thông điệp $m' \in \mathbb{F}_2^{k_1 k_2} = (\mathbb{F}_2^{k_2})^{k_1}$, gọi $(x_1, x_2, ..., x_n) = C_{out}(m')$, trong đó $x_i \in \mathbb{F}_2^{k_2}$, ta có:

$$C_{out} \circ (C_{in}^1, ..., C_{in}^{n_1})(m') = (C_{in}^1(x_1), ..., C_{in}^{n_1}(x_{n_1})). \quad (2.6)$$

Hay nói cách khác, mỗi ký tự của mã ngoài được thay bằng mã tự trong tương ứng. Mã trong đơn giản nhất là mã đơn vị $I_q : \mathbb{F}_2^{k_2} \to \mathbb{F}_2^{2^{k_2}}$, trong đó mỗi thành viên của $\mathbb{F}_2^{k_2}$ được ánh xạ 1-1 đến một vector đơn vị khác nhau của $\mathbb{F}_2^{2^{k_2}}$.

Xét C_{out} là mã RS với các tham số $[n_1, k_1, n_1 - k_1 + 1]_q$ và tất cả các mã trong đều là mã đơn vị I_q. Đặt các từ mã của mã nối vào các cột của ma trận *M*. Ma trận

này có kích thước $n_1 2^{k_2} \times 2^{k_1 k_2}$. Mỗi cột của ma trận có trọng số đúng bằng n_1. Khoảng cách Hamming giữa hai cột khác nhau ít nhất là $n_1 - k_1 + 1$, do đó phần chung giữa hai cột có kích thước nhiều nhất là $k_1 - 1$. Khi đó, *M* là ma trận d-phân-cách với giá trị *d* được xác định như sau:

$$d = \left\lfloor \frac{n_1 - 1}{k_1 - 1} \right\rfloor \tag{2.7}$$

Với phương pháp này, ta có $t = O(d^2 \log^2 N)$. Từ điều kiện $n_1 \le q$ của mã Reed Solomon và công thức (2.7), luận án chọn giá trị n_1 và giá trị k_1 nhỏ để có được giá trị lớn nhất của *d*. Giá trị *d* thể hiện số lượng Hot-IP tối đa mà giải pháp có thể phát hiện được, giá trị này lựa chọn tùy thuộc vào các ứng dụng thực tế. Như vậy, luận án chọn mã ngoài C_{out} là mã RS $[q-1, k]_q$ và C_{in} là ma trận đơn vị I_q để phát sinh ma trận nhị phân d-phân-cách trong các thực nghiệm của giải pháp. Ma trận *M* thu được từ phép nối mã $C_{out} \circ C_{in}$ bằng cách thay thế $N = q^k$ từ mã trong C_{out} bằng các cột trong ma trận.

Ví dụ về phương pháp nối mã với C_{out}: $[q, k]_q$, C_{in}: $[q] \rightarrow \{0, 1\}^q$, cho *k=1*, *q=3*, C_{out}=(0, 0, 0), (1, 1, 1), (2, 2, 2), (0, 1, 2), (1, 2, 0), (2, 0, 1) và $C_{in}(j)$=(00…j…00) với giá trị 1 nằm ở vị trí thứ j. Ta có kết quả của phép nối mã như sau:

$$C_{out}: \begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 \\ 0 & 1 & 2 & 2 & 0 & 1 \end{bmatrix} \quad C_{in}: \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C_{out} \circ C_{in}: \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

❖ **Điều kiện áp dụng phương pháp thử nhóm đạt hiệu quả**

Định lý 1: (***Chặn Bassalygo***) Gọi $t(d,N)$ là số hàng nhỏ nhất của ma trận d-phân-cách với *N* cột, ta có:

$$t(d,N) \geq \min\left\{\binom{d+2}{2}, N\right\} \quad (2.8)$$

Như vậy, nếu d quá lớn thỏa $(d+1)(d+2) \geq 2N$ thì thử nhóm không tốt hơn thử các mẫu đơn lẻ. Định lý này được chứng minh trong [55].

2.4. PHÁT HIỆN HOT-IP DÙNG THỬ NHÓM BẤT ỨNG BIẾN

2.4.1. Phát biểu bài toán

Bài toán xác định các Hot-IP sử dụng phương pháp thử nhóm bất ứng biến được phát biểu như sau:

Cho một dòng *m* gói IP có địa chỉ tương ứng S=(IP_1, IP_2,…,IP_m), với *m* rất lớn. Mỗi gói tin IP có địa chỉ IP trong tập [N], N cũng rất lớn (N=2^{32} với IPv4, N=2^{128} với IPv6). Gọi $f_i = \left|\{j \middle| IP_i = IP_j; i \neq j; IP_i, IP_j \in S\}\right|$, thì

$$\text{Hot-IP} = \{IP_i \in S \,|\, f_i \geq \phi \times m, 0 \leq \phi \leq 1\}.$$

Giả sử có đối đa *d* Hot-IP trong dòng gói tin IP. Xác định các Hot-IP trong *S*.

Bài toán phát hiện các Hot-IP có thể giải bằng phương pháp thử nhóm bất ứng biến được mô hình hóa như sau: cho trước ma trận nhị phân $M_{t\times N}$ với *t* là hàm phụ thuộc d và N. Trong đó, *t* là số hàng của ma trận tương ứng với các nhóm thử trong thử nhóm bất ứng biến, N là số cột của ma trận tương ứng với N địa chỉ IP phân biệt và M là ma trận nhị phân d-phân-cách được phát sinh bằng phương pháp nối mã. Gọi m_{ij} là phần tử của ma trận ở hàng i, cột j; các phần tử của ma trận có giá trị như sau:

$$m_{ij} = \begin{cases} 1 & \text{nếu } IP_j \text{ thuộc nhóm thử i} \\ 0 & \text{ngược lại} \end{cases}$$

Giả sử có vector kết quả $r_{t\times 1}$ sau khi cập nhật địa chỉ IP trong các gói tin từ dòng dữ liệu và xét ngưỡng, các r_i có giá trị như sau:

$$r_i = \begin{cases} 1 & \text{nếu nhóm thử có chứa Hot-IP} \\ 0 & \text{ngược lại} \end{cases}$$

Ta cần xác định xem những IP nào là Hot-IP.

2.4.2. Giải pháp phát hiện các Hot-IP

Phương pháp giải bài toán phát hiện các Hot-IP dựa vào phương pháp thử nhóm bất ứng biến truyền thống được tóm tắt như sau:

Ma trận nhị phân M d-phân-cách được xác định trước, sử dụng *t* bộ đếm $c_1, c_2, ..., c_t$ tương ứng với số dòng của ma trận nhị phân M d-phân-cách, khi một gói tin có địa chỉ IP $j \in [N]$ tới thì tăng tất cả các bộ đếm c_i nếu $m_{ij} = 1$. Từ các bộ đếm này và một ngưỡng cho trước, một vector kết quả được tạo ra $r \in \{0,1\}^t$. Trong đó, kết quả của các nhóm thử có chứa Hot-IP là 1 và kết quả của các nhóm thử không chứa Hot-IP là 0. Các phần tử của r được xác định như sau:

$$r_i = \begin{cases} 1 & \text{nếu } c_i \geq m/(d+1) \\ 0 & \text{ngược lại} \end{cases}$$

Giá trị ngưỡng dùng để xác định kết quả nhóm thử như trên được nhóm tác giả Cormode đề xuất trong [27], thuật toán khởi tạo và tính toán vector kết quả được trình bày trong thuật toán 1 "Khởi tạo và tính toán vector kết quả".

Giá trị *m* trong giải pháp phát hiện Hot-IP trực tuyến có thể xác định tùy vào năng lực của thiết bị triển khai giải pháp trong một chu kỳ thuật toán, cụ thể giá trị này có thể ước lượng dựa vào khả năng xử lý số lượng gói tin trong khoảng thời gian một chu kỳ thuật toán.

Giá trị *N* cũng có thể xem xét dựa vào năng lực xử lý của vị trí triển khai hoặc có thể xác định dựa vào ứng dụng cụ thể. Các tham số này được trình bày trong các phần tiếp theo.

Thuật toán 1	***Khởi tạo và tính toán vector kết quả***

Input:

- *M là ma trận d-phân-cách có kích thước* $t \times N$
- $C := (c_1,...,c_t) \in N^t$
- $r:=(r_1,...,r_t) \in \{0,1\}^t$
- *S: dãy các địa chỉ IP trong dòng gói tin IP*

Output: *Vector kết quả R*

```
1:  For t=1 to t do c_i=0
2:  For each j ∈ S
3:      For i=1 to t do
4:          If m_ij=1 then c_i++
5:      End For
6:  End For
7:  For i=1 to t do
8:      If c_i ≥ m/(d+1) then
9:           r_i=1
10:     Else
11:         r_i=0
12:     End If
13: End For
```

❖ **Xác định các Hot-IP trong dòng dữ liệu:**

Gọi $x = (x_1, x_2, ..., x_n) \in \{0, 1\}^n$, với $x_i = 1$ khi và chỉ khi IP *i* là Hot-IP, ngược lại thì $x_i=0$. Gọi $r = (r_1, r_2, ..., r_t) \in \{0,1\}^t$, với $r_i = 1$ khi và chỉ khi $c_i \geq \frac{m}{d+1}$, ngược lại thì $r_i=0$. Gọi T là tập các Hot-IP, vector kết quả chính là hội của các cột của M tương ứng với T, $r_i = \vee_{j \in T_i} x_j$. Ta có $Mx = r$ và $|x| \leq d$. Xác định các *x* chính là các Hot-IP cần tìm.

Xác định *x* được tóm tắt như sau: với dữ liệu đầu vào là địa chỉ IP được trích ra trong dòng các gói tin, ma trận d-phân-cách *M* và vector kết quả $r \in \{0,1\}^t$,

$x_j = 1,\ \forall j \in [N]$. Xét từng nhóm thử, nếu $r_i = 0,\ \forall j \in [N]$, nếu $m_{ij} = 1$ thì gán $x_j = 0$.

$$\begin{array}{c} 1\ \ 2\ \ 3\ \ \ldots\ldots\ \ N \\ \begin{bmatrix} 1 & 0 & 0 & \ldots\ldots & 1 \\ 0 & 0 & 1 & \ldots\ldots & 0 \\ 0 & 0 & 0 & \ldots\ldots & 1 \\ & & & . & \\ & & & . & \\ & & & . & \\ 1 & 1 & 1 & \ldots\ldots & 0 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ . \\ . \\ . \\ t \end{array} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ . \\ . \\ . \\ r_t \end{bmatrix}$$

Thuật toán xác định các Hot-IP trong dòng gói IP được mô tả trong thuật toán 2 "Xác định các Hot-IP". Xét các nhóm thử có kết quả "âm tính" (r_i=0), các nhóm thử không chứa Hot-IP, loại bỏ các IP thuộc các nhóm này. Sau khi xem xét và các IP tương ứng trong các nhóm này, những địa chỉ IP còn lại là các Hot-IP.

Thuật toán 2	***Xác định các Hot-IP*** (thuật toán giải mã đơn giản)
Input:	*Ma trận nhị phân M d-phân-cách và vector kết quả r*
Output:	*Các Hot-IP*

```
1: With each ri=0 do
2:   For i=1 To N do
3:       If (mij=1) then
4:           IP:=IP\{j}
5:       Endif
6:   End For
7: End With
8: Return IP  //tập các IP còn lại
```

Ví dụ 1: Cho dòng gói tin IP, địa chỉ IP được trích ra trong IP-header được ánh xạ thành dãy giá trị các số như sau IP={1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5}, ma trận 2-phân-cách với các nhóm thử được thiết kế như sau:

$$M_{9\times7}=\begin{bmatrix}0&0&1&0&0&1&1\\0&0&1&0&1&0&0\\0&0&1&1&0&0&0\\0&1&0&0&0&1&0\\0&1&0&0&1&0&1\\0&1&0&1&0&0&0\\1&0&0&0&0&1&0\\1&0&0&0&1&0&0\\1&0&0&1&0&0&1\end{bmatrix}$$

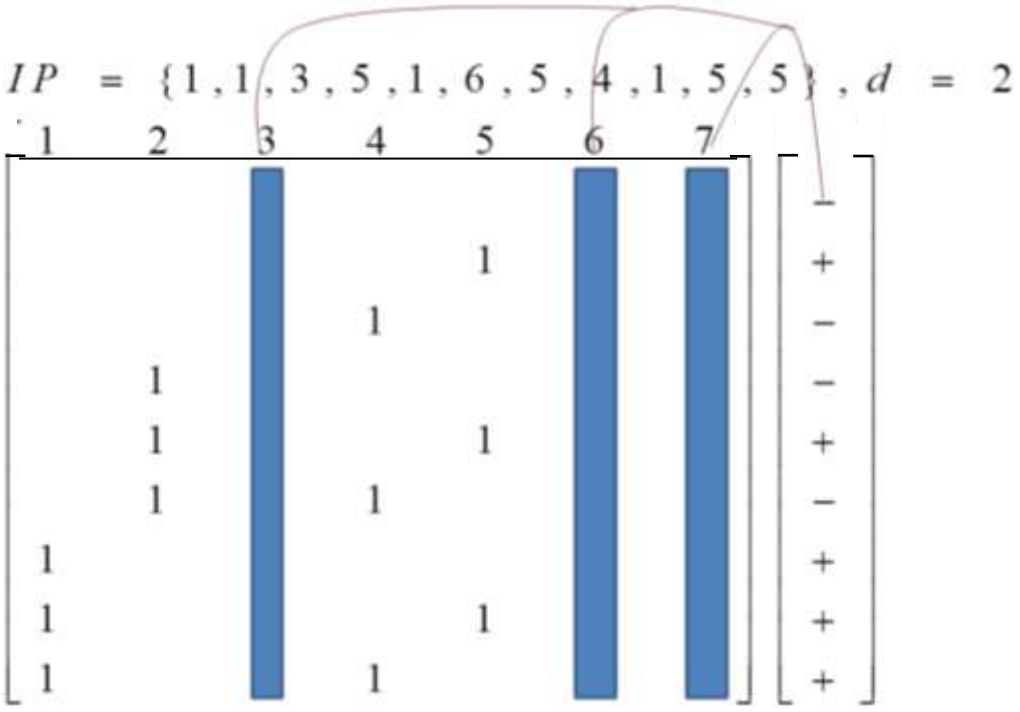
Tính toán các bộ đếm và vector kết quả như dựa vào dãy các phần tử đầu vào, ta có vector bộ đếm c_1=2, c_2=5, c_3=2, c_4=1, c_5=4, c_6=1, c_7=5, c_8=8, c_9=5. Một nhóm thử chứa phần tử là Hot-IP nếu bộ đếm của phép thử đó lớn hơn (tổng số gói/(d+1))=11/(2+1) theo cách tính ngưỡng được đề xuất trong [39]. Từ đó, chúng ta suy ra được vector kết quả như sau: r_1=0, r_2=1, r_3=0, r_4=0, r_5=1, r_6=0, r_7=1, r_8=1, r_9=1. Trong hình 2.2. dấu "–" tương ứng với r_i=0 và dấu "+" tương ứng với r_i=1.

$$IP = \{1,1,3,5,1,6,5,4,1,5,5\},\ d = 2$$

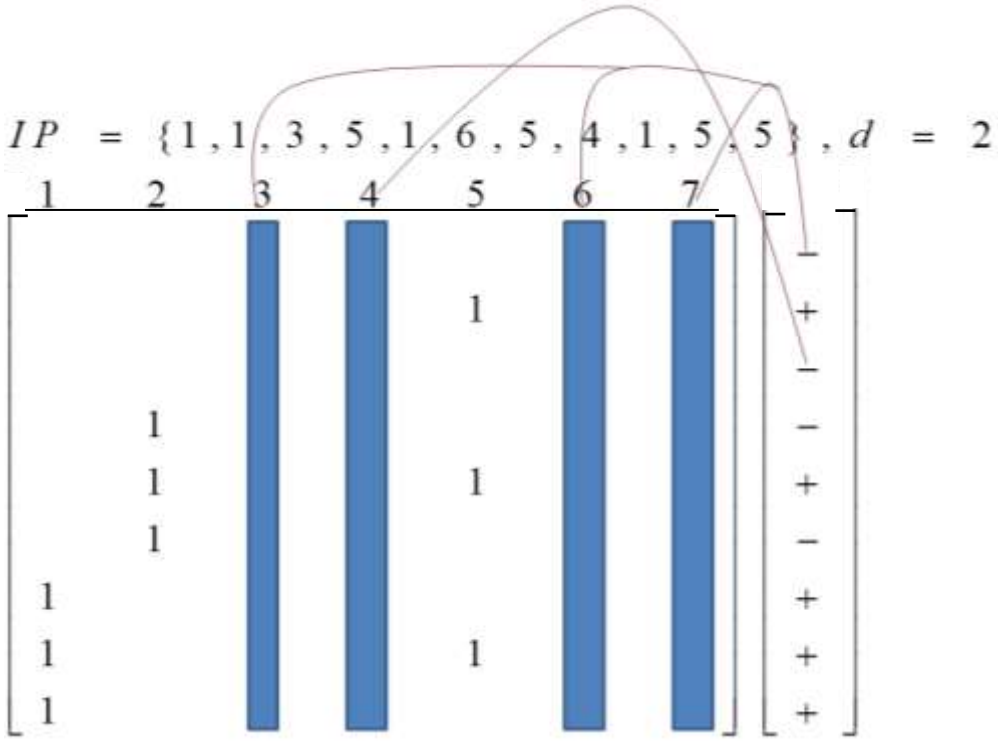
IP	1	2	3	4	5	6	7	
2			1			1	1	–
5			1		1			+
2			1	1				–
1		1				1		–
4		1			1		1	+
1		1		1				–
5	1					1		+
8	1				1			+
5	1			1			1	+

Hình 2.2. *Ví dụ về giải mã phát hiện các Hot-IP*

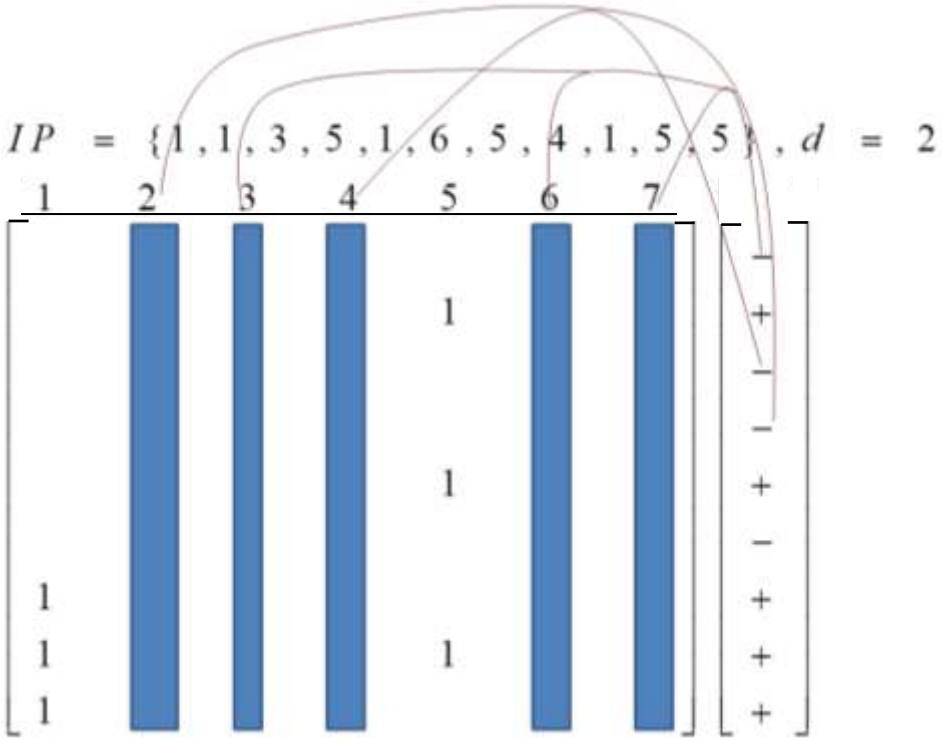
Sơ lược các bước thực hiện tìm Hot-IP được mô tả trong các hình 2.3, hình 2.4 và hình 2.5. Trong đó, xem xét các nhóm thử có kết quả "âm tính" để tiến hành loại bỏ các IP tương ứng thuộc các nhóm này.



Hình 2.3. *Loại các cột j tương ứng với $m_{1j}=1$ và $r_1=0$*



Hình 2.4. *Loại các cột j tương ứng với $m_{3j}=1$ và $r_3=0$*



Hình 2.5. *Loại các cột j tương ứng với $m_{4j}=1$ và $r_4=0$*

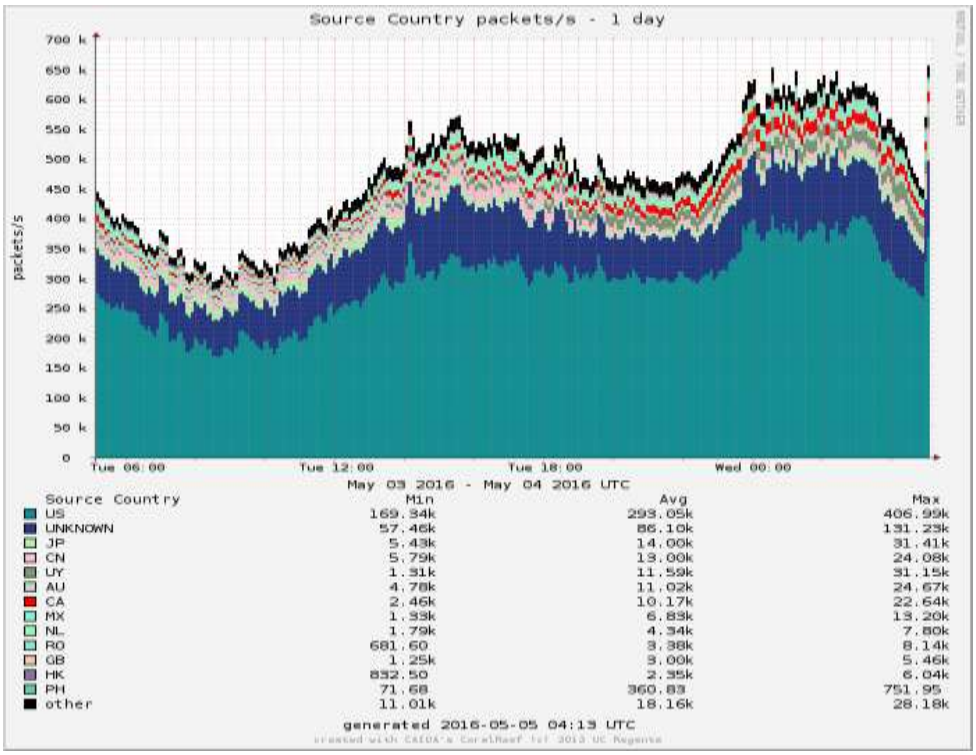
Kết quả: Hot-IP là IP thứ 1và 5.

2.4.3. Những vấn đề nghiên cứu đặt ra

Bài toán phát hiện các Hot-IP là bài toán có ý nghĩa quan trọng trong an ninh mạng, đặc biệt ở các mạng trung gian như mạng của các nhà cung cấp dịch vụ (ISP), với số lượng gói tin xử lý qua các thiết bị và tần suất truy cập rất lớn, để triển khai giải pháp phát hiện trực tuyến cần phải xem xét các yếu tố ảnh hưởng tới việc xử lý của thuật toán từ việc nhận dữ liệu đầu vào, lưu trữ và xử lý để giảm thời gian tính toán là những yếu tố quan trọng.

Trong kiến trúc tổng quát của một ISP, bên cạnh các kết nối từ phía mạng khách hàng đến ISP, còn có các kết nối từ phía ISP đến các ISP khác. Với vai trò là môi trường mạng trung gian xử lý và chuyển tiếp các gói tin từ nguồn đến đích, số lượng gói tin và số lượng địa chỉ IP qua các thiết bị định tuyến là rất lớn. Theo số liệu công bố từ trung tâm phân tích ứng dụng dữ liệu Internet CAIDA [68]; lưu lượng Internet được thu thập từ router đặt ở Chicago kết nối với hệ thống mạng lõi

Tier 1-ISP giữa Chicago (IL) và Seattle (WA); số lượng gói tin đi qua router ở thời điểm đo thể hiện ở hình 2.6.



Hình 2.6. *Số lượng gói tin qua router và phân loại theo nguồn* [70]

Dữ liệu thu thập được từ router của một ISP ở New Zealand của nhóm nghiên cứu WAND [73] (Đại học Waikado – New Zealand) ở một số thời điểm năm 2010 được thể hiện trên bảng 2.2.

Nhóm nghiên cứu về đo lường và phân tích dữ liệu Internet MAWI [69] (Nhật Bản) công bố lưu lượng mạng chuyển tiếp qua mạng lõi WIDE được thu thập trong khoảng 15 phút mỗi lần. Bảng 2.3 trình bày số liệu về số lượng gói tin và số lượng địa chỉ IP ở một số thời điểm đo của nhóm MAWI.

Qua các số liệu trên cho thấy rằng lưu lượng mạng, số lượng địa chỉ IP qua router ở các mạng trung gian như các ISP đang rất lớn và ngày càng tăng nhanh. Theo dự báo về lưu lượng Internet từ hãng Cisco thì lưu lượng mạng và số lượng địa chỉ IP qua các ISP ngày một lớn hơn rất nhiều, các thiết bị IoT đang bắt đầu phát triển nhanh và bùng nổ trong thời gian ngắn sắp tới [74]. Theo đó, dự báo về sự phát triển của lưu lượng Internet toàn cầu đến năm 2020 sẽ tăng gần gấp 3 lần so

với hiện tại, lưu lượng giờ nghẽn sẽ tăng theo hệ số 4,6 giữa năm 2015 và 2020, lưu lượng Internet trung bình sẽ tăng theo hệ số 2,0. Số lượng thiết bị kết nối vào Internet sẽ tăng gấp 3 lần so với dân số toàn cầu vào năm 2020.

Bảng 2.2. *Số lượng địa chỉ IP qua router của một ISP ở New Zealand [72]*

Ngày	**Thời gian** (giờ/phút/giây)	**Số gói tin**	**Số lượng IP phân biệt**
06/01/2010	16:09:46 - 16:30:00	23.636.605	502.298
07/01/2010	03:00:01 - 03:30:00	12.423.587	373.906
09/01/2010	04:00:01 - 04:30:00	13.236.581	414.095
13/01/2010	09:30:01 - 10:00:01	24.407.988	423.484
17/01/2010	01:00:01 - 01:30:01	18.781.458	473.965
18/01/2010	08:30:01 - 09:00:01	20.549.673	462.233

Bảng 2.3. *Số lượng gói tin và địa chỉ IP đi qua mạng lõi chuyển tiếp WIDE* [71]

Ngày	**Thời gian** (giờ/phút/giây)	**Số gói tin**	**Số lượng IP phân biệt**
30/03/2012	00:00:00 - 00:15:00	37.661.325	1.494.673
	06:00:00 - 06:15:01	30.823.712	1.358.883
	12:00:00 - 12:15:00	30.238.356	1.549.983
	19:00:01 - 19:15:01	41.267.930	1.541.106
	23:00:01 - 23:15:00	38.304.965	1.527.405
01/04/2012	01:45:00- 02:00:00	26.494.277	1.553.361
	10:30:00 - 10:45:00	20.969.854	1.350.974
02/10/2014	00:00:01 - 00:15:01	87.184.982	22.114.023
	14:15:01 - 14:30:01	123.476.984	34.995.759
	18:00:17 - 18:15:24	118.889.505	30.581.292
	21:00:02 - 21:15:02	102.590.713	27.676.532
	23:00:01 - 23:15:00	118.661.810	32.264.683
	23:30:01 - 23:45:01	109.676.534	35.241.088

Cùng với sự mở rộng băng thông mạng, tốc độ trên cổng vật lý của các thiết bị mạng được nâng cấp, một số giải pháp để kiểm soát lưu lượng mạng như giải pháp phát hiện các Hot-IP có vai trò quan trọng nhằm giúp theo dõi, cảnh báo, hạn chế hay ngăn chặn các đối tượng (Hot-IP) có khả năng ảnh hưởng đến hoạt động ổn định của toàn hệ thống.

Giải pháp phát hiện nhanh các Hot-IP trên mạng với mục tiêu chính là phát hiện các IP có số lượng gói tin xuất hiện rất lớn trong khoảng thời gian rất ngắn. Các ứng dụng chính của giải pháp này như phát hiện các đối tượng có khả năng là sâu đang quét mạng (dạng sâu quét không gian địa chỉ IP – một số dạng *"scanning worm"* như *"hit-list worm"* hay *"routing worm"*) nhằm phát hiện lỗ hổng của các thiết bị trên mạng để lây nhiễm, phát hiện các đối tượng có khả năng là nguồn phát hay mục tiêu trong các tấn công từ chối dịch vụ (DoS/DDoS) với các nguồn phát liên tục số lượng gói tin rất lớn làm "tràn ngập" gây quá tải hệ thống của mục tiêu tấn công. Như vậy, giải pháp phát hiện nhanh các Hot-IP trên mạng được đề xuất trong luận án tập trung giải quyết bài toán phát hiện các IP xuất hiện tần suất cao trên mạng, một số ứng dụng từ việc phát hiện các Hot-IP được trình bày chi tiết trong chương 4 của luận án.

Đối với việc phân bố tần suất xuất hiện các IP phân biệt, từ dữ liệu thực tế thu thập từ router ở một ISP của nhóm WAND trong thời gian 30 phút, phân bố tần suất xuất hiện của các IP phân biệt được thể hiện ở bảng 2.4. Trong đó, tổng số IP phân biệt trong dữ liệu thu thập được trong 30 phút là 305.454, số IP được trích ra và thể hiện trong bảng phân bố tần suất là 304.431. Qua bảng phân bố tần suất này cho thấy tần suất xuất hiện các IP có số lượng gói nhỏ hơn 50 (gói tin) chiếm tỷ lệ rất lớn (98,3576%), các địa chỉ IP có số lượng gói tin xuất hiện lớn hơn 5.000 (gói tin) chiếm tỷ lệ rất nhỏ trong tập dữ liệu thu thập được, không thể hiện trong bảng phân bố tần suất này [73]. Một số dự liệu thực tế khác từ nhóm nghiên cứu MAWI cũng cho thấy các IP xuất hiện với tần suất cao (số lượng gói tin lớn) chiếm tỉ lệ rất nhỏ so với các IP bình thường (tần suất thấp) chiếm tỷ lệ rất lớn trong lưu lượng dữ liệu Internet.

Bảng 2.4 . *Phân bố tần suất xuất hiện của các IP phân biệt từ dữ liệu nhóm WAND.*

Số lượng gói tin	Số IP	Tần suất	Số lượng gói tin	Số IP	Tần suất
1-50	299.431	98,3576%	601-650	74	0,0243%
51-100	1.671	0,5489%	651-700	61	0,0200%
101-150	799	0,2625%	701-750	70	0,0230%
151-200	518	0,1702%	751-800	44	0,0145%
201-250	387	0,1271%	801-850	48	0,0158%
251-300	242	0,0795%	851-900	47	0,0154%
301-350	215	0,0706%	901-950	37	0,0122%
351-400	182	0,0598%	951-1.000	41	0,0135%
401-450	146	0,0480%	1.001-1.050	36	0,0118%
451-500	113	0,0371%	1.051-1.100	33	0,0108%
501-550	97	0,0319%	1.101-1.150	35	0,0115%
551-600	74	0,0243%	1.151-1.200	30	0,0099%

Mặc dù phương pháp thử nhóm bất ứng biến có nhiều ưu điểm hơn các thuật toán như đã trình bày ở phần trước về phương diện tính toán nhanh hơn, tính đơn giản của giải pháp và mức độ chính xác cao. Để ứng dụng giải pháp vào việc phát hiện trực tuyến các Hot-IP trên mạng đạt hiệu quả cần phải xem xét, cải tiến các yếu tố sau:

(1) Lựa chọn kích thước của ma trận d-phân-cách phù hợp với vị trí triển khai và khả năng của hệ thống (xác định t và N). Vấn đề này có thể xem xét ở hai trường hợp ứng dụng. Thứ nhất, nếu ứng dụng trong các mạng trung gian ở các ISP, các IP được xem xét như nhau thì việc lựa chọn giá trị N theo năng lực xử lý trên thiết bị cài đặt trong một chu kỳ thuật toán. Thứ hai, nếu ứng dụng trên các mạng cung cấp dịch vụ ngoài Internet cho người dùng, khi đó có sự phân biệt IP của những người dùng đăng ký sử dụng dịch vụ và những IP không đăng ký. Khi đó có thể xác định giá trị N dựa trên các IP đăng ký sử dụng dịch vụ và một số địa chỉ IP đại diện cho các IP không đăng ký.

(2) Xác định số lượng Hot-IP tối đa cho giải pháp. Trong phương pháp thử nhóm bất ứng biến truyền thống giá trị d là tham số ước lượng cho trước, thể hiện số lượng Hot-IP tối đa mà giải pháp có thể phát hiện được. Việc lựa chọn d lớn ảnh hưởng đến việc tăng kích thước ma trận trong phương pháp nối mã để đảm bảo ma trận d-phân-cách. Để giải quyết vấn đền này, một danh sách lưu các IP nghi ngờ (có khả năng là Hot-IP) được sử dụng trong quá trình thực thi thuật toán, danh sách chứa các IP nghi ngờ này có thể được mở rộng kích thước, giúp giảm sự phụ thuộc vào giá trị *d* trong ma trận d-phân-cách.

(3) Xác định ngưỡng tần suất cao như thế nào cho phù hợp. Trong phương pháp thử nhóm bất ứng biến truyền thống ứng dụng trong xác định phần tử tần suất cao, giá trị ngưỡng được xác định phụ thuộc vào số lượng phần tử trên dòng dữ liệu xem xét và ma trận d-phân-cách. Để ứng dụng trong việc phát hiện trực tuyến trong một chu kỳ thuật toán, có thể xém xét khả năng của thiết bị triển khai nhận được số lượng gói tin tối đa trong một chu kỳ thuật toán và kích thước của danh sách IP nghi ngờ.

(4) Cải tiến thuật toán thử nhóm bất ứng biến để tăng hiệu quả tính toán, độ chính xác và phát hiện trực tuyến.

(5) Nâng cao hiệu quả phát hiện Hot-IP bằng một số kỹ thuật kết hợp như xử lý song song, kiến trúc phân tán trong việc triển khai giải pháp ở các hệ thống mạng được tổ chức đa vùng.

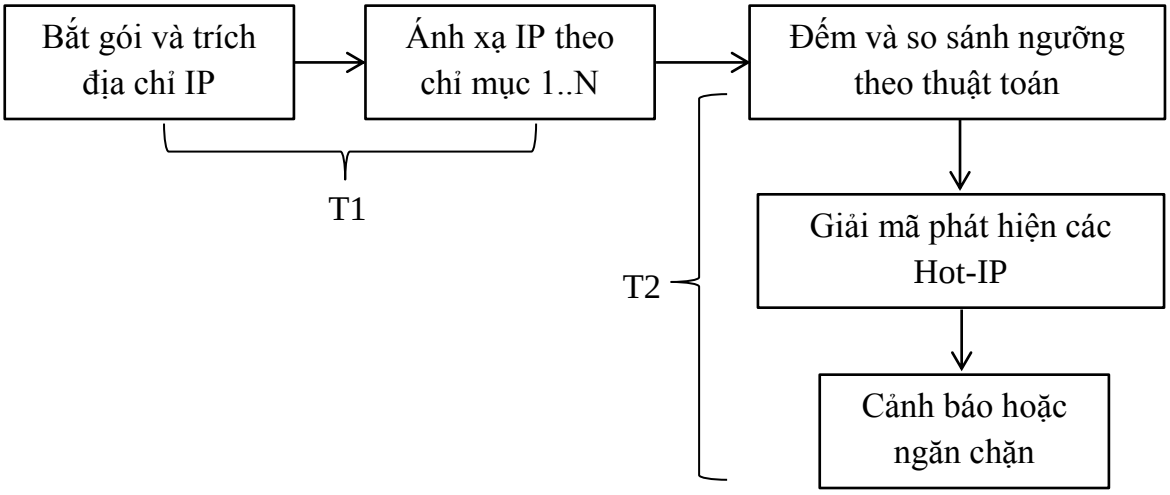
2.5. ĐỀ XUẤT THUẬT TOÁN CẢI TIẾN

Các phân tích ở chương 1 về các nghiên cứu liên quan cho thấy phương pháp "*counter-based*" thực thi rất nhanh trong trường hợp áp dụng cho số lượng phần tử nhỏ. Trong phần này, luận án trình bày thuật toán cải tiến để phát hiện nhanh hơn và chính xác hơn các Hot-IP trực tuyến trên mạng bằng cách kết hợp phương pháp thử nhóm bất ứng biến với phương pháp *"counter-based"*.

Thuật toán cải tiến áp dụng vào việc tính toán trực tuyến được thiết lập ở khu vực biên mạng để phát hiện, ngăn chặn các Hot-IP nhằm giúp hệ thống mạng hoạt

động ổn định và giúp người quản trị theo dõi các khả năng có nguy cơ ảnh hưởng đến hoạt động của mạng và dịch vụ. Ngoài ra, có thể triển khai giải pháp ở phía các mạng trung gian ở các nhà cung cấp dịch vụ để phát hiện sớm các nguy cơ ảnh hưởng đến hệ thống mạng của mạng khách hàng. Tiến trình của giải pháp phát hiện các Hot-IP trên mạng qua hai giai đoạn như sau:

- Giai đoạn khởi tạo: người quản trị xác định các tham số cho hệ thống gồm kích thước ma trận, ngưỡng tần suất cao, chu kỳ thực hiện thuật toán theo khả năng của vị trí triển khai, sinh ma trận và tải ma trận vào hệ thống.
- Giai đoạn phát hiện Hot-IP trực tuyến: tiến trình thực hiện giải pháp được thể hiện sau đây. Tổng thời gian thực hiện chương trình: T = T1 + T2. Các giai đoạn thực hiện giải pháp được thể hiện trên hình 2.7.



Hình 2.7. *Tiến trình thực hiện giải pháp*

Ý tưởng chính cho thuật toán cải tiến phương pháp thử nhóm bất ứng biến trong bài toán phát hiện các Hot-IP này là:

(1) Việc cập nhật các bộ đếm khi một IP đến cho từng nhóm sẽ dừng lại nếu nó vượt ngưỡng.

(2) Xác định các IP làm vượt ngưỡng trong nhóm này, đưa vào danh sách nghi ngờ và thiết lập bộ đếm tương ứng.

(3) Nếu một IP đến có trong danh sách nghi ngờ thì tăng bộ đếm tương ứng cho IP đó mà không cập nhật các bộ đếm trong các nhóm thử chứa địa chỉ IP này.

(4) Xác định Hot-IP bằng cách so sánh bộ đếm của các IP trong danh sách nghi ngờ với ngưỡng tần suất cao.

2.5.1. Thuật toán cải tiến 1 – "Online Hot-IP Detecting"

Thuật toán cải tiến 1: *Online Hot-IP Detecting*	
	Input: Ma trận d-phân-cách, dòng gói tin IP trong chu kỳ ***Output:*** các Hot-IP
1:	`Hot-List={}`
2:	*For each IP* $j \in S_\Delta$ *// đối với mỗi gói tin IP đến*
3:	*If (current_timestamp- reference_timestamp <* Δ *) then*
4:	*If IP j* $\in$ *Hot-List then*
5:	*Hot-List[j].count++*
6:	*Else*
7:	*For i = 1 to N*
8:	*If* m_{ij} *= 1 and* $c_i < \delta$ *then* c_i*++*
9:	*If* $c_i \geq \delta$ *then*
10:	*Hot-List = Hot-List* $\cup$ *{j}*
11:	*Hot-List[j].count = min{*c_i *|* m_{ij}*=1}*
12:	*EndIf*
13:	*EndFor*
14:	*Else*
15:	*Return* $\{j \mid \text{Hot-List[j].count} \geq \delta, 1 \leq j \leq \mid \text{Hot-List} \mid\}$
16:	*//xuất ra các IP trong Hot-List có bộ đếm tương ứng vượt ngưỡng*
17:	*Reference_timestamp=current_timestamp*
18:	*Reset Hot-List*
19:	*EndIf*

Thuật toán cải tiến 1 "Online Hot-IP Detecting" thực hiện theo dõi các gói tin trực tuyến và xuất các Hot-IP phát hiện được trong một chu kỳ thuật toán. Trong

chu kỳ thực hiện thuật toán, các gói tin được trích địa chỉ IP và thực hiện việc cập nhật trong danh sách địa chỉ IP nghi ngờ (Hot-List) hay trong các bộ đếm của các nhóm thử.

Khi một địa chỉ IP được trích ra từ gói tin IP đến, nó sẽ được kiểm tra trong danh sách Hot-List, nếu tồn tại trong danh sách này thì tăng bộ đếm tương ứng cho IP này. Nếu chưa tồn tại trong Hot-List thì việc cập nhật cho các nhóm thử chứa IP này được thực hiện bình thường như trong thuật toán thử nhóm bất ứng biến truyền thống.

Khi bất kỳ một nhóm nào trong quá trình cập nhật IP mới vào làm vượt ngưỡng tần suất cao, địa chỉ IP đó được đưa vào danh sách nghi ngờ Hot-List, khởi tạo bộ đếm tương ứng bằng cách lấy giá trị nhỏ nhất trong các nhóm mà IP này thuộc về, các nhóm vượt ngưỡng sẽ dừng việc cập nhật.

Trong thuật toán cải tiến 1, gọi "current_timestamp" là thời gian các gói tin đến, "reference_timestamp" là điểm bắt đầu của chu kỳ thuật toán, Δ là thời gian một chu kỳ thuật toán.

Ví dụ 2: Cho dòng gói tin IP, các địa chỉ IP được trích ra trong IP-header là dòng liên tục các IP (IP stream) như sau IP={1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5}, ma trận 2-phân-cách với các nhóm thử được thiết kế như ở ví dụ 1, ngưỡng $\delta = 3$.

$$IP = \{1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5\},\ d = 2$$

	IP	1	2	3	4	5	6	7	
C_1	2			1			1	1	−
C_2	5			1		1			+
C_3	2			1	1				−
C_4	1		1				1		−
C_5	4		1			1		1	+
C_6	1		1		1				−
C_7	5	1					1		+
C_8	8	1				1			+
C_9	5	1			1			1	+

Khởi tạo Hot-List{}

Dòng IP: 1 (hệ thống nhận được địa chỉ IP 1, nó cập nhật vector C)

	1	2	3	4	5	6	7	8	9
C							1	1	1

Dòng IP: 1, 1

	1	2	3	4	5	6	7	8	9
C							2	2	2

Dòng IP: 1, 1, 3

	1	2	3	4	5	6	7	8	9
C	1	1	1				2	2	2

Dòng IP: 1, 1, 3, 5

	1	2	3	4	5	6	7	8	9
C	1	2	1		1		2	3	2

C[8] đến ngưỡng, đưa IP 5 vào Hot-List và khởi tạo bộ đếm cho IP 5 là min{C_j với j=1..5 và m_{j5}=1}=min{C_2, C_5, C_8}=1. Ta có Hot-List {(5, 1)}, y=1.

Dòng IP: 1, 1, 3, 5, 1

	1	2	3	4	5	6	7	8	9
C	1	2	1		1		3	3	3

C_7 đến ngưỡng, đưa IP 1 vào Hot-List và khởi tạo bộ đếm cho IP 1 là min{C_j với j=1..t và m_{j1}=1}=min{C_7, C_8, C_9}=3. Ta có Hot-List {(5, 1), (1, 3)}, y=2.

Dòng IP: 1, 1, 3, 5, 1, 6

	1	2	3	4	5	6	7	8	9
C	2	2	1	1	1		3	3	3

Dòng IP: 1, 1, 3, 5, 1, 6, 5

	1	2	3	4	5	6	7	8	9
C	2	2	1	1	1		3	3	3

Cập nhật bộ đếm trong Hot-List {(5, 2), (1, 3)}, y=2.

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4

	1	2	3	4	5	6	7	8	9
C	2	2	2	1	1	1	3	3	3

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1

	1	2	3	4	5	6	7	8	9
C	2	2	2	1	1	1	3	3	3

Cập nhật bộ đếm trong Hot-List {(5, 2), (1, 4)}, y=2.

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5

Cập nhật bộ đếm trong Hot-List {(5, 3), (1, 4)}, y=2.

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5

Cập nhật bộ đếm trong Hot-List {(5, 4), (1, 4)}, y=2.

Kết quả tìm được Hot-IP là IP: 5 và 1.

- ***Kịch bản thử nghiệm so sánh thời gian giải mã của phương pháp thử nhóm bất ứng biến truyền thống và thử nhóm bất ứng biến cải tiến:***

Mục tiêu của thử nghiệm này để đo khả năng xử lý của giải pháp cải tiến với giải pháp trước với số lượng IP phân biệt khác nhau. Trong thử nghiệm này, các gói tin được phát sinh ngẫu nhiên bởi chương trình. Các IP đóng vai trò là IP thông thường được phát sinh với tỉ lệ xuất hiện ngẫu nhiên nhỏ (5 – 100 gói), các IP đóng vai trò là Hot-IP được phát sinh với tần suất xuất hiện lớn (5.000 – 100.000 gói). Cơ sở để lựa chọn các giá trị này như sau: Trong khoảng thời gian chu kỳ thuật toán trong các thử nghiệm (5 giây, 10 giây, 15 giây, 20 giây, 25 giây, 30 giây), các máy tính đại diện cho IP thông thường thực hiện các truy cập đến các mục tiêu bằng lệnh "ping" và truy cập các trang web của mục tiêu. Bằng công cụ phân tích gói Wireshark có thể đo được số lượng nhiều nhất trong các lần thực nghiệm của máy tính đại diện cho IP bình thường nhỏ hơn 100 gói tin. Đối với các Hot-IP, trong thử nghiệm sử dụng các công cụ tấn công như DoSHTTP, Trinoo để phát sinh gói tin lớn, số lượng gói tin có thể điều chỉnh ở nhiều mức và đã sử dụng ở mức từ 5.000 –

100.000 gói trong thời gian chu kỳ thuật toán. Việc lựa chọn các thông số thử nghiệm dựa trên cơ sở các công cụ thử nghiệm như vậy. Bên cạnh đó, một số tham khảo khác từ các nghiên cứu lưu lượng tấn công từ chối dịch vụ của CAIDA [75] cho thấy việc sử dụng ngưỡng cho các IP tần suất cao ở mức 5.000 là hợp lý làm căn cứ cho các lựa chọn bên trên trong thử nghiệm cho giải pháp đề xuất.

Số lượng IP phân biệt sử dụng để đo trong thử nghiệm với các mức khác nhau từ 3.000 – 260.000 địa chỉ (thể hiện ở bảng 2.5). Thời gian chạy thuật toán được đo theo thời gian hệ thống (sử dụng hàm thời gian trong chương trình, tính từ thời gian bắt đầu chạy thuật toán đến khi kết thúc thuật toán). Trong trường hợp này chỉ xét đến thời gian giải mã của giải thuật, chưa tính đến thời gian bắt gói và xử lý gói tin để trích ra địa chỉ IP. Thử nghiệm so sánh thời gian giải mã được thực hiện trên server có cấu hình: IBM Xeon E5420 2.5 GHz, RAM 4GB, hệ điều hành CentOS 64 bit. Kết quả thực nghiệm được trình bày trong bảng 2.5.

Bảng 2.5. *Thời gian giải mã của thuật toán thử nhóm và thuật toán cải tiến*

N *(đ/v 1000)*	**GT** *(giây)*	**GT cải tiến** *(giây)*	**Chênh lệch** *(giây)*	**N** *(đ/v 1000)*	**GT** *(giây)*	**GT cải tiến** *(giây)*	**Chênh lệch** *(giây)*
3	0,08	0,05	0,03	100	2,28	1,29	0,99
5	0,14	0,09	0,05	120	2,79	1,55	1,24
7	0,16	0,11	0,05	140	3,19	1,82	1,37
9	0,21	0,14	0,07	160	3,65	2,08	1,57
11	0,26	0,15	0,11	180	4,10	2,34	1,76
20	0,48	0,26	0,22	200	4,56	2,61	1,95
40	1,01	0,53	0,48	220	5,01	2,88	2,13
60	1,37	0,80	0,57	240	5,48	3,14	2,34
80	1,84	1,04	0,80	260	5,93	3,39	2,54

Qua phần thực nghiệm giải mã của hai phương pháp cho thấy rằng thuật toán giải mã với phương pháp thử nhóm bất ứng biến cải tiến cho kết quả tốt hơn. Với danh sách Hot-List, việc cập nhật sẽ được tiến hành nhanh hơn rất nhiều so với cập nhật các nhóm thử phải tra trong ma trận d-phân-cách để xác định các nhóm thử cần cập nhật.

- ***Trường hợp số lượng Hot-IP trong dòng gói tin IP lớn hơn giá trị d trong ma trận d-phân-cách:***

Với trường hợp trên dòng gói tin IP có nhiều hơn *d* Hot-IP sẽ có khả năng phát hiện sai đối với phương pháp thử nhóm bất ứng biến truyền thống vì khi đó xuất hiện nhiều nhóm thử mang giá trị không chính xác gọi là dương tính giả. Khi đó, với thuật toán giải mã đơn giản sẽ phát hiện nhiều giá trị không phải là Hot-IP.

Trong trường hợp số lượng Hot-IP thực sự lớn hơn giá trị *d*, thuật toán thử nhóm bất ứng biến cải tiến cho kết quả chính xác hơn thuật toán thử nhóm bất ứng biến truyền thống. Tính đúng đắn của thử nghiệm dùng thuật toán giải mã đơn giản phụ thuộc vào độ chính xác của giá trị dự đoán *d* của người thiết kế: nếu *d* nhỏ, tính đúng đắn thấp do vấn đề dương tính giả ảnh hưởng đến quá trình cập nhật các nhóm thử. Nếu *d* lớn, tính đúng đắn cao hơn, nhưng bù lại phải trả giá về độ phức tạp (cả tính toán lẫn lưu trữ).

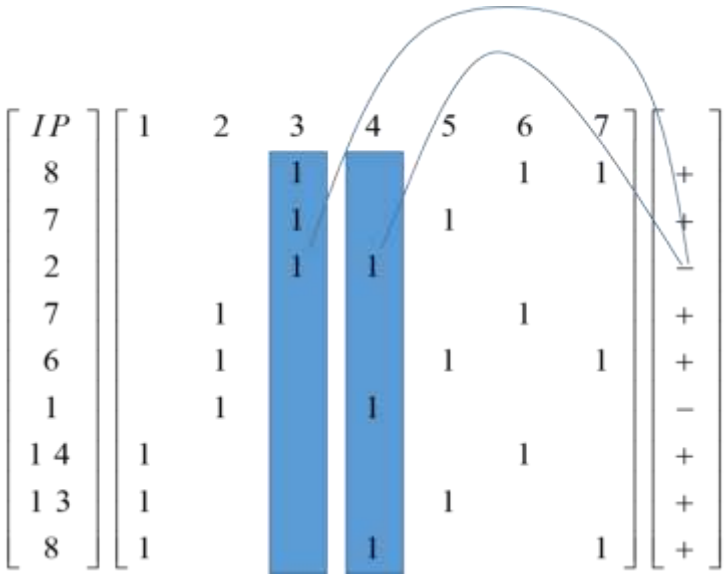
Do đó, việc chọn danh sách IP nghi ngờ Hot-List là sự dung hòa giữa chi phí và tính đúng đắn. Thuật toán thử nhóm bất ứng biến cải tiến chính là sự kết hợp của phương pháp thử nhóm bất ứng biến (phương pháp dựa vào *sketch*) và phương pháp "*counter-based"* với số lượng bộ đếm nhỏ, việc cập nhật cho các phần tử nghi ngờ sẽ cập nhật trực tiếp trên danh sách nhỏ này mà không phải xem xét trên toàn ma trận *d*-phân-cách. Đồng thời, thuật toán cải tiến cho phép mở rộng số phần tử trong danh sách nghi ngờ. Do đó, thuật toán thử nhóm bất ứng biến cải tiến cho kết quả chính xác hơn do có thể mở rộng kích thước của danh sách chứa đựng các IP nghi ngờ mà không ảnh hưởng đến việc thay đổi kích thước của ma trận d-phân-cách. Thuật toán cải tiến giải quyết vấn đề giảm thời gian tính toán và độ chính xác do

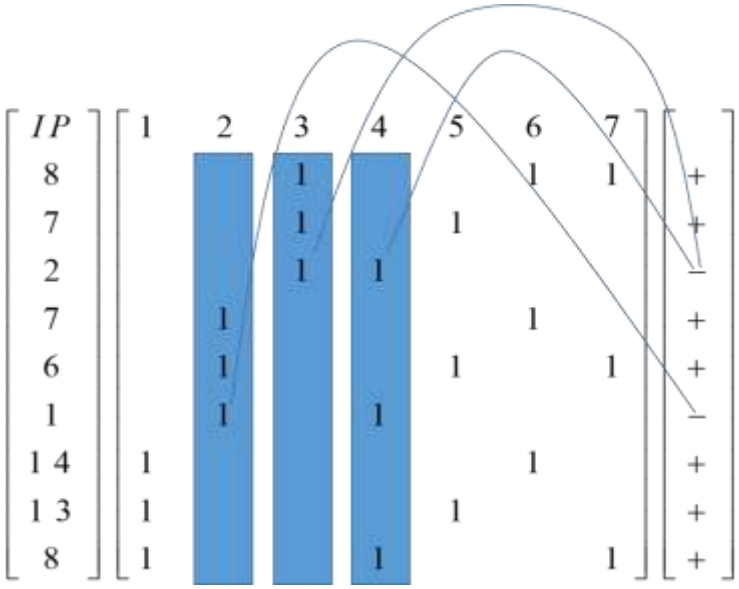
cập nhật trên danh sách nhỏ hơn, do vậy có thể chọn kích thước Hot-List lớn, giảm sự phụ thuộc vào *d*.

Sau đây là ví dụ minh họa cho trường hợp số Hot-IP thực tế lớn hơn giá trị *d* cho trước với hai thuật toán: thử nhóm bất ứng biến truyền thống trong ví dụ 3 và thử nhóm bất ứng biến cải tiến trong ví dụ 4.

Ví dụ 3: Cho dòng gói tin IP, địa chỉ IP được trích ra trong IP-header được ánh xạ thành dãy giá trị các số như sau IP={1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 1, 1, 5, 5, 6, 6, 6, 6, 6, 6}, ma trận 2-phân-cách với các nhóm thử được thiết kế như ở ví dụ 1, ngưỡng $\delta = 5$.

	IP	1	2	3	4	5	6	7		
C_1	8			1			1	1	+	r_1
C_2	7			1		1			+	r_2
C_3	2			1	1				−	r_3
C_4	7		1				1		+	r_4
C_5	6		1			1		1	+	r_5
C_6	1		1		1				−	r_6
C_7	14	1					1		+	r_7
C_8	13	1				1			+	r_8
C_9	8	1			1			1	+	r_9





Hot-IP phát hiện được: 1, 5, 6, 7. Hot-IP thật sự là: 1, 5, 6.

Ví dụ 4: Chạy lại ví dụ 3 dùng phương pháp thử nhóm bất ứng biến cải tiến. Dòng dữ liệu IP={1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 1, 1, 5, 5, 6, 6, 6, 6, 6, 6}, ma trận 2-phân-cách với các nhóm thử được thiết kế như ở ví dụ 1, ngưỡng $\delta = 5$.

Dòng IP: 1

	1	2	3	4	5	6	7	8	9
C							1	1	1

Dòng IP: 1, 1

	1	2	3	4	5	6	7	8	9
C							2	2	2

Dòng IP: 1, 1, 3

	1	2	3	4	5	6	7	8	9
C	1	1	1				2	2	2

Dòng IP: 1, 1, 3, 5

	1	2	3	4	5	6	7	8	9
C	1	2	1		1		2	3	2

Dòng IP: 1, 1, 3, 5, 1

	1	2	3	4	5	6	7	8	9
C	1	2	1		1		3	4	3

Dòng IP: 1, 1, 3, 5, 1, 6

	1	2	3	4	5	6	7	8	9
C	2	2	1	1	1		4	4	3

Dòng IP: 1, 1, 3, 5, 1, 6, 5

	1	2	3	4	5	6	7	8	9
C	2	3	1	1	2		4	5	3

C_8 đến ngưỡng, đặt IP 5 và Hot-List = {(5, min{c_2, c_5, c_8})} = {(5, 2)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4

	1	2	3	4	5	6	7	8	9
C	2	3	2	1	2	1	4	5	4

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1

	1	2	3	4	5	6	7	8	9
C	2	3	2	1	2	1	5	5	5

C_7 và C_9 tới ngưỡng, đưa IP 1 vào Hot-List = {(5, 2), (1, 5)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5

IP 5 có trong Hot-List, cập nhật bộ đếm cho IP 5. Hot-List = {(5, 3), (1, 5)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5

IP 5 có trong Hot-List, cập nhật bộ đếm cho IP 5. Hot-List = {(5, 4), (1, 5)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1

IP 1 có trong Hot-List, cập nhật bộ đếm cho IP 1. Hot-List = {(5, 4), (1, 6)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 1

IP 1 có trong Hot-List, cập nhật bộ đếm cho IP 1. Hot-List = {(5, 4), (1, 7)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5

IP 5 có trong Hot-List, cập nhật bộ đếm cho IP 5. Hot-List = {(5, 5), (1, 6)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5

IP 5 có trong Hot-List, cập nhật bộ đếm cho IP 5. Hot-List = {(5, 6), (1, 6)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6

	1	2	3	4	5	6	7	8	9
C	3	3	2	2	2	1	5	5	5

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6, 6

	1	2	3	4	5	6	7	8	9
C	4	3	2	3	2	1	5	5	5

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6, 6, 6

	1	2	3	4	5	6	7	8	9
C	5	3	2	4	2	1	5	5	5

C_1 tới ngưỡng, đưa IP 6 vào Hot-List = {(5, 6), (1, 6), (6, 4)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6, 6, 6, 6

IP 6 có trong Hot-List, cập nhật bộ đếm cho IP 6. Hot-List = {(5, 6), (1, 6), (6, 5)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6, 6, 6, 6, 6

IP 6 có trong Hot-List, cập nhật bộ đếm cho IP 6. Hot-List = {(5, 6), (1, 6), (6, 6)}

Dòng IP: 1, 1, 3, 5, 1, 6, 5, 4, 1, 5, 5, 1, 5, 5, 6, 6, 6, 6, 6, 6

IP 6 có trong Hot-List, cập nhật bộ đếm cho IP 6. Hot-List = {(5, 6), (1, 6), (6, 7)}

Hot-IP tìm được là IP: 1, 5, 6

- **Thực nghiệm so sánh độ chính xác của thử nhóm bất ứng biến truyền thống và thuật toán cải tiến 1 "Online Hot-IP Detecting"**

Trong phần thực nghiệm này, tác giả mô phỏng tấn công từ chối dịch vụ với công cụ tấn công Trinoo và DoSHTTP có cường độ tấn công khác nhau. Thời gian của thuật toán được tính cả thời gian bắt gói, xử lý gói và thời gian tính toán tìm ra các Hot-IP.

Mô hình thực nghiệm gồm 01 máy Web server, 20 máy phát tấn công, 01 máy cài đặt giải pháp phát hiện Hot-IP đặt trước Web server và 35 máy đóng vai trò là các máy tính của người dùng bình thường. Hệ điều hành sử dụng trong các máy thử nghiệm là CentOS 6.2. Giải pháp được cài đặt bằng ngôn ngữ lập trình C, máy cài đặt giải pháp đóng vai trò là điểm giám sát để phát hiện các IP là Hot-IP đến Web server. Luồng lưu lượng được thu thập qua cổng mạng của máy cài đặt giải pháp theo chiều vào, các IP nguồn trong các gói tin được trích ra từ IP-header được sử dụng làm tham số đầu vào trong thuật toán.

Kịch bản thực nghiệm so sánh mức độ giải mã chính xác tìm các Hot-IP trên mạng với số lượng Hot-IP thực tế lớn hơn giá trị *d* trong ma trận d-phân-cách. Tham số sử dụng: ma trận d-phân-cách 240x4096, d=7, |Hot-List| = 14. Sử dụng phần mềm Trinoo tấn công DDoS vào máy chủ nạn nhân với lần lượt số lượng các máy tấn công là: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20. Mỗi giá trị làm thực nghiệm 5 lần, lấy kết quả nhỏ nhất. Kết quả thực nghiệm được trình bày trong bảng 2.6.

Các máy giả lập là máy tấn công thực hiện phát số lượng gói tin lớn (Hot-IP) sử dụng công cụ DoSHTTP và Trinoo để tiến hành tấn công vào Web server. Các máy tính đóng vai trò là các máy tính của người dùng bình thường, thực hiện các lệnh "ping" và truy cập vào website của Web server sử dụng trình duyệt Web thông thường. Chu kỳ giải thuật được thử nghiệm với thời gian 5 giây, 10 giây, 15 giây, 20 giây, 25 giây và 30 giây.

Qua các phân tích và thực nghiệm cho thấy thuật toán cải tiến có nhiều ưu điểm hơn phương pháp thử nhóm bất ứng biến truyền thống. Thứ nhất là không cập nhật các nhóm thử đã đến ngưỡng, thứ hai là thay vì cập nhật một IP trong dòng dữ liệu cho tất cả các nhóm thử chứa IP đó thì chỉ cần cập nhật trong danh sách nghi

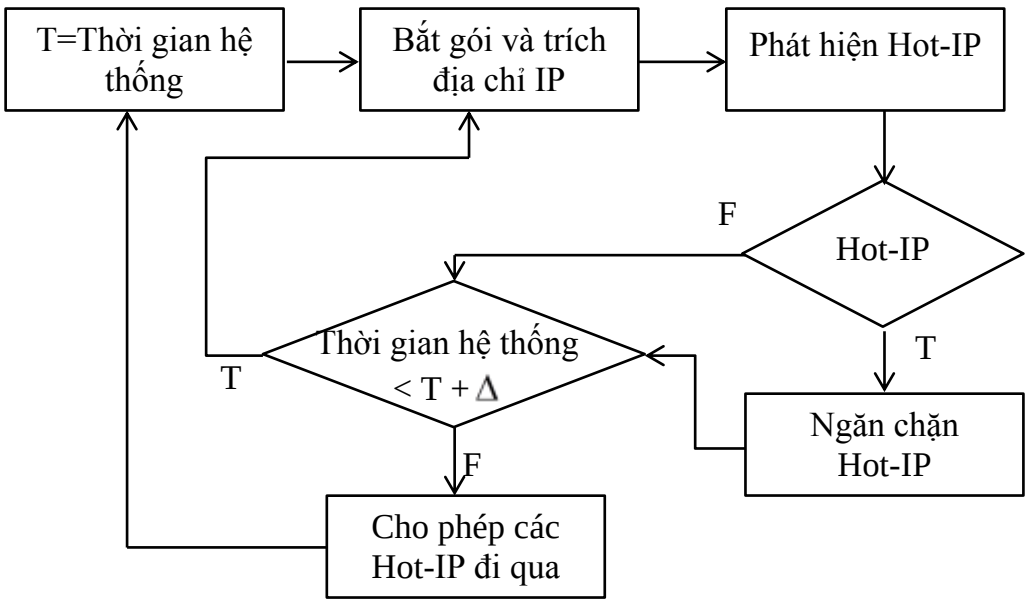
ngờ. Đồng thời, thuật toán cải tiến này cho kết quả chính xác hơn trong trường hợp số lượng Hot-IP thực tế nhiều hơn *d* (số Hot-IP tối đa có thể phát hiện được trong thử nhóm bất ứng biến truyền thống).

***Bảng 2.6**. So sánh độ chính xác của thử nhóm bất ứng biến truyền thống và cải tiến*

Hot-IP thực tế	D	Kích thước Hot-List	GT truyền thống (độ chính xác)	GT cải tiến 1 (độ chính xác)
2	7	14	100%	100%
4	7	14	100%	100%
6	7	14	100%	100%
8	7	14	100%	100%
10	7	14	95%	100%
12	7	14	95%	100%
14	7	14	92%	100%
16	7	14	90%	100%
18	7	14	90%	98%
20	7	14	90%	98%

2.5.2. Thuật toán cải tiến 2 – "Online Hot-IP Preventing"

Lưu đồ thuật toán như sau:



***Hình 2.8.** Lưu đồ giải pháp hạn chế ảnh hưởng của các Hot-IP*

Thuật toán cải tiến 2: *Online Hot-IP Preventing*

Input: Ma trận nhị phân d-phân-cách, dòng gói tin IP, ngưỡng δ

Δ: chu kỳ thuật toán

Xử lý:

```
1:  T=Systemtime + Δ,
2:     Khởi tạo:cho phép tất cả các IP đi qua
3:  For each IP j đến và (Systemtime <T)
4:    j=get(IP)
5:    if IP j ∈ Hot-List then
6:          Hot-List[j].count++
7:          If Hot-List[j].count > δ then drop(IP j)
8:    Else
9:          Cập nhật các bộ đếm c_i với m_ij=1,
10:         không cập nhật cho các c_i vượt ngưỡng
11:       If c_i > δ then
12:         Đưa IP j vào Hot-List
13:         Khởi tạo bộ đếm cho IP j = min{c_i với m_ij=1}
14:       endIf
15:   EndIf
16: EndFor
```

Trên cơ sở thuật toán cải tiến 1 *"Online Hot-IP Detecting"* dùng để phát hiện các Hot-IP trực tuyến bằng việc sử dụng danh sách IP nghi ngờ, việc cập nhật và tính toán được cải thiện tốc độ tính toán. Thuật toán cải tiến 2 *"Online Hot-IP Preventing"* thực hiện ngăn chặn các IP có khả năng là nguy cơ ngay khi chúng được phát hiện bằng cách ngăn chặn các Hot-IP trong một chu kỳ thời gian của thuật toán nhằm đảm bảo hệ thống hoạt động ổn định, thông suốt.

Thuật toán cải tiến 2 có thể ứng dụng ở các mạng ISP hoặc IsSP (Internet special Service Provider). Trong mạng ISP các IP được xem xét như nhau, trong

mạng cung cấp một dịch vụ nào đó cho người dùng trên Internet (IsSP) có sự phân biệt người dùng (IP) đăng ký sử dụng dịch vụ và người dùng không đăng ký.

Đối với người dùng đăng ký sử dụng dịch vụ, khả năng ngắt kết nối ít hơn. Người dùng không đăng ký, khả năng bị ngắt kết nối cao hơn do khả năng xuất hiện Hot-IP cao hơn vì những người dùng này sẽ được ánh xạ vào chung một số ít địa chỉ IP đại diện.

❖ Kịch bản thực nghiệm thuật toán cải tiến 2 *"Online Hot-IP Preventing"*:

Mô hình thực nghiệm gồm 01 Web server, 03 máy phát tấn công, 01 máy cài đặt giải pháp được đặt trước Web server và 35 máy tính đóng vai trò là các máy tính của người dùng bình thường. Hệ điều hành sử dụng trên máy cài đặt giải pháp là CentOS, giải pháp được cài đặt bằng ngôn ngữ lập trình C. Luồng lưu lượng truy cập tới Web server được kiểm soát bởi máy cài đặt giải pháp phát hiện Hot-IP, các gói tin được thu thập qua cổng mạng của máy này để xử lý trong chương trình.

Các máy tấn công sử dụng công cụ Trinoo để tiến hành phát lưu lượng tấn công vào Web server. Các máy tính đóng vai trò là các máy tính của người dùng bình thường thực hiện các truy cập vào website của Web server. Chu kỳ thuật hiện thuật toán là 15 giây.

Phần thực hiện mô phỏng tấn công được lặp lại với nhiều tình huống tấn công khác nhau và cường độ tấn công khác nhau, kết quả ghi nhận được trong các trường hợp này thể hiện ở mức độ đáp ứng của Web server khi bị tác động của tấn công trước và sau khi cài đặt giải pháp.

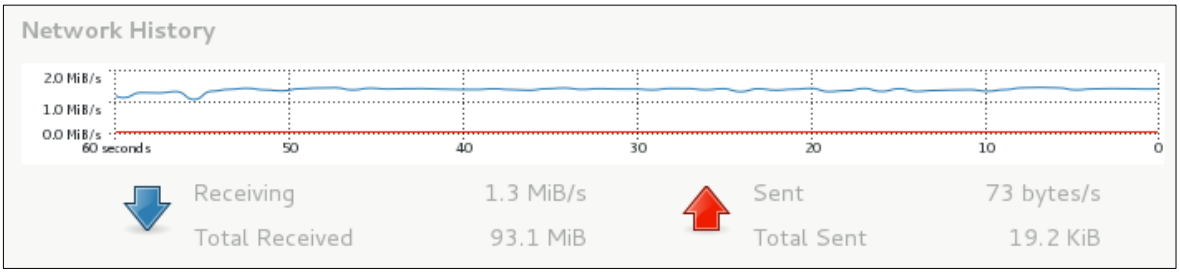
Các tham số sử dụng trong thực nghiệm:

- Ma trận nhị phân M 7-phân-cách: 240x4096
- Chu kỳ thực hiện thuật toán $\Delta = 15$ giây
- Năng lực hệ thống ở vị trí triển khai giải pháp là số lượng gói tin tối đa có thể xử lý được trong một chu kỳ thực hiện thuật toán (m_Δ).

- Ngưỡng $\delta = \frac{m_{\Delta}}{|\text{Hot-List}|}$. Trong thực nghiệm lựa chọn |Hot-List| =14.

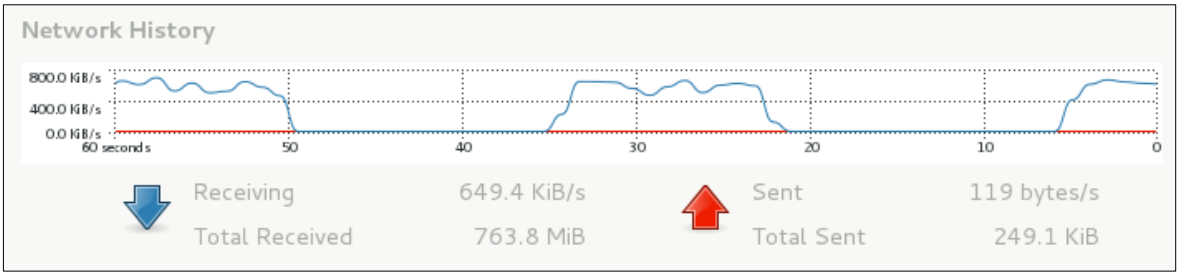
Kết quả thực nghiệm:

- Trước khi cài đặt giải pháp: khi bị tấn công từ chối dịch vụ với công cụ Trinoo, lưu lượng trên máy chủ nạn được thể hiện trên hình 2.9.



Hình 2.9. *Các thông số hệ thống của máy chủ khi bị tấn công DDoS*

- Sau khi cài đặt giải pháp: các thông số trên máy chủ nạn nhân hoạt động ổn định do các Hot-IP đã bị phát hiện và ngăn chặn không cho vào hệ thống trong một chu kỳ thuật toán. Hình 2.10 thể hiện các thông số máy chủ nạn nhân khi sử dụng giải pháp ngăn chặn các Hot-IP trong một chu kỳ thuật toán.



Hình 2.10. *Các thông số của máy chủ khi cài giải pháp ngăn chặn Hot-IP*

Các Hot-IP được phát hiện và bị ngăn chặn trong một chu kỳ thuật toán thể hiện trên hình 2.11.

```
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.1.11         anywhere
DROP       all  --  192.168.1.5          anywhere
```

Hình 2.11. *Các Hot-IP bị khóa trong một chu kỳ thuật toán*

Thuật toán cải tiến 2 *"Online Hot-IP Preventing"* có thể dùng để triển khai ở các router biên trước các máy chủ cung cấp dịch vụ hoặc ở các router trung gian trong các mạng trung gian đáp ứng mục tiêu đảm bảo cho hệ thống mạng hoạt động ổn định, thông suốt.

2.6. KẾT LUẬN CHƯƠNG 2

Trong chương này, luận án trình bày một số đặc điểm cơ bản của phương pháp thử nhóm bất ứng biến, một số khái niệm liên quan, phương pháp xây dựng tường minh ma trận d-phân-cách bằng phép nối mã, mô hình hóa bài toán phát hiện Hot-IP bằng phương pháp thử nhóm bất ứng biến, thuật toán phát hiện các Hot-IP sử dụng thuật toán giải mã đơn giản. Bên cạnh đó, luận án đề xuất hai thuật toán cải tiến phương pháp thử nhóm bất ứng biến để đáp ứng việc phát hiện và ngăn chặn trực tuyến các Hot-IP là "Online Hot-IP Detecting" và "Online Hot-IP Preventing".

Phương pháp xây dựng d-phân-cách dựa vào xác suất mặc dù cho kết quả số dòng nhỏ hơn nhưng có điểm yếu trong cách xây dựng, mức độ chính xác và lưu trữ khi xử lý ma trận này cần phải lưu trữ toàn bộ trong bộ nhớ khi thực thi chương trình. Do đó, cách xây dựng ma trận bằng xác suất không hiệu quả khi triển khai vào ứng dụng thực tế trên các thiết bị mạng mà vốn tài nguyên hệ thống có hạn và cần tối ưu về không gian lưu trữ. Hơn nữa, việc phát sinh ma trận ma trận theo phương pháp xác suất có khả năng phát sinh ma trận không phải là d-phân-cách không sẽ ảnh hưởng đến kết quả giải mã chính xác của thuật toán.

Phương pháp nối mã được áp dụng để xây dựng ma trận d-phân-cách tường minh và chính xác từ mã Reed-Solomon và mã đơn vị thực hiện đơn giản và có thể phát sinh các cột khi xử lý mà không cần phải lưu toàn bộ ma trận. Đây là đặc điểm giúp giải quyết tốt việc tối ưu không gian lưu trữ ma trận có kích thước lớn để triển khai giải pháp trên các thiết bị có tài nguyên hạn chế.

Luận án đã mô hình hóa bài toán phát hiện Hot-IP về bài toán thử nhóm bất ứng biến và thuật toán phát hiện Hot-IP trên mạng dựa trên ma trận d-phân-cách và vector kết quả của các nhóm thử. Bên cạnh đó, luận án trình bày một số khái niệm

liên quan đến ý tưởng xây dựng ma trận d-phân-cách-danh-sách giải mã nhanh của nhóm Indyk-Ngo-Rudra. Hạn chế của phương pháp này là vấn đề xây dựng ma trận phân cách và phân cách danh sách dẫn đến tốn bộ nhớ, xác suất không cao. Hai thuật toán cải tiến phương pháp thử nhóm bất biến được đề xuất là "Online Hot-IP detecting" và "Online Hot-IP preventing" cải thiện về khả năng tính toán, độ chính xác và đảm bảo hệ thống hoạt động ổn định, thông suốt. Các nội dung chính của chương này được công bố ở các công trình [C1][C4][C7] trong danh mục các công trình nghiên cứu của tác giả.

Bài toán phát hiện Hot-IP trong dòng gói tin IP trên mạng cần được xem xét thêm một số khía cạnh khác để có thể áp dụng triển khai vào thực tế như lựa chọn kích thước ma trận phân cách theo vị trí triển khai, các bước tính toán xử lý phải được tối ưu về thời gian với các kỹ thuật kết hợp như xử lý song song, kiến trúc phân tán trong các hệ thống mạng đa vùng. Những vấn đề này sẽ được trình bày trong chương 3.

CHƯƠNG 3. NÂNG CAO HIỆU QUẢ PHÁT HIỆN HOT-IP BẰNG MỘT SỐ KỸ THUẬT KẾT HỢP

3.1. GIỚI THIỆU

Để triển khai giải pháp phát hiện Hot-IP trên mạng ở một vị trí cụ thể cần có những phân tích để tối ưu tính toán. Một số vấn đề cần quan tâm xem xét là mô hình triển khai tập trung hay phân tán, khả năng xử lý song song của giải pháp, số lượng địa chỉ IP cần giám sát để lựa chọn kích thước ma trận phù hợp với vị trí triển khai.

Trong chương 2, luận án đã trình bày mô hình hóa bài toán phát hiện Hot-IP trên mạng dùng phương pháp thử nhóm bất ứng biến, phương pháp giải, hai thuật toán cải tiến để nâng cao hiệu quả tính toán và áp dụng vào phát hiện các Hot-IP trực tuyến. Chương này trình bày một số kỹ thuật kết hợp nhằm nâng cao hiệu quả phát hiện nhanh các Hot-IP trên mạng để có thể áp dụng triển khai trên các mạng tốc độ cao ở các nhà cung cấp dịch vụ.

Một số kỹ thuật có thể kết hợp để nâng cao khả năng của giải pháp trong việc phát hiện nhanh các Hot-IP như: (i) lựa chọn kích thước của ma trận d-phân-cách phù hợp để giảm thời gian và không gian xử lý dựa vào khả năng của vị trí triển khai giải pháp, (ii) sử dụng kỹ thuật xử lý song song để nâng cao khả năng tính toán và (iii) sử dụng kiến trúc phân tán để tổ chức triển khai giải pháp ở các khu vực và cảnh báo sớm đến các khu vực khác trong các hệ thống mạng đa vùng.

Dựa vào vị trí triển khai cụ thể có thể xác định được hai tham số quan trọng: tham số thứ nhất là thời gian một chu kỳ thực hiện thuật toán (Δ), tham số này có ý nghĩa là khoảng thời gian trước khi mất kết nối hay thời gian chỉ mức độ chịu đựng của hệ thống; tham số thứ hai là ngưỡng tần suất cao (δ), tham số này có ý nghĩa là khả năng tiếp nhận số lượng gói tin trên dòng gói tin IP, được tính toán dựa trên băng thông đường truyền và năng lực xử lý của server cung cấp dịch vụ tại vị trí triển khai cụ thể. Đây là hai tham số cố định trong bài toán phát hiện các Hot-IP

được xác định tại vị trí triển khai. Như vậy, các tham số khác sẽ được phân tích cụ thể dựa trên sự cố định của hai tham số này theo ý nghĩa như trên.

3.2. VẤN ĐỀ KÍCH THƯỚC MA TRẬN PHÂN CÁCH

Việc lựa chọn kích thước của ma trận d-phân-cách có ý nghĩa quan trọng để áp dụng vào thực tế có hiệu quả. Kích thước ma trận ảnh hưởng đến thời gian cập nhật các gói dữ liệu trong dòng dữ liệu đầu vào và thời gian thực hiện thuật toán để phát hiện ra các Hot-IP một cách đáng kể.

Thời gian giải mã của phương pháp thử nhóm để tìm ra các Hot-IP là $O(tN)$ như đã đề cập trong chương 2. Một cách hiển nhiên có thể thấy rằng kích thước ma trận lớn, nghĩa là số lượng IP giám sát lớn và số hàng của ma trận lớn sẽ làm tăng thời gian cập nhật và tính toán. Trong phần này trình bày 2 nội dung: nội dung thứ nhất là thử nghiệm việc giải mã với kích thước ma trận khác nhau để thấy được mức độ ảnh hưởng của kích thước ma trận đến thời gian giải mã tìm ra các Hot-IP; nội dung thứ hai trình bày một số căn cứ để chọn lựa các tham số cho ma trận nhằm thực hiện hiệu quả giải pháp trong việc ứng dụng tại các vị trí triển khai cụ thể.

3.2.1. Sự ảnh hưởng của kích thước ma trận

Luận án tiến hành đo thời gian giải mã để phát hiện Hot-IP với số lượng địa chỉ cho trước khác nhau và kích thước ma trận khác nhau. Mục đích của việc này để thấy được mức độ ảnh hưởng của kích thước ma trận, số lượng phần tử tham gia vào quá trình giải mã của giải pháp.

Trong phần thử nghiệm, các gói tin chứa các Hot-IP được phát sinh ngẫu nhiên. Cài đặt thực nghiệm trên Server IBM (Xeon E5420 2.5 GHz, RAM 4GB, hệ điều hành CentOS 6.4 (64 bit)), thời gian thu thập gói tin trong dòng IP được bỏ qua, kích thước của ma trận d-phân-cách được sử dụng khác nhau. Từ đó tính toán thời gian giải mã, tức là thời gian chạy thuật toán để tìm ra các Hot-IP. Kết quả giải mã tìm ra các Hot-IP trong dòng gói tin IP được trình bày trong các bảng 3.1, 3.2, 3.3 và 3.4. Bảng 3.1 là kết quả thực nghiệm với các ma trận có kích thước nhỏ.

Bảng 3.1. *Thời gian giải mã với kích thước ma trận khác nhau*

RS code	t	D	Thời gian *(giây)*	N (IP)
$[15,3]_{16}$	240	7	0,11	4.096
$[31,3]_{32}$	992	15	3,65	32.768

Bảng 3.2 trình bày kết quả thời gian giải mã với kích thước ma trận từ phép nối mã sử dụng RS-$[31,5]_{32}$, ma trận này có khả năng xử lý đến 33.554.432 IP phân biệt cùng lúc. Ma trận *M* sử dụng thực nghiệm có kích thước cố định 992x1000000, số lượng IP trong dòng gói tin IP thay đổi từ 100.000 đến 1.000.000 địa chỉ (d=7, t=992).

Bảng 3.2. *Thời gian giải mã với ma trận con xây dựng từ RS-*$[31,5]_{32}$

N (IP)	Thời gian giải mã (giây)	N (IP)	Thời gian giải mã (giây)
100.000	0,66	600.000	4,46
200.000	1,34	800.000	6,25
400.000	2,78	1.000.000	8,16

Ma trận 240x1.048.576 với *d=3* được sinh ra từ mã RS-$[15,5]_{16}$, thực nghiệm lần lượt với các N={30.000, 200.000, 400.000, 600.000, 800.000, 1.000.000} được trình bày trong Bảng 3.3.

Bảng 3.3. *Thời gian giải mã với ma trận xây dựng từ RS-*$[15,5]_{16}$.

N (IP)	Thời gian giải mã (giây)	N (IP)	Thời gian giải mã (giây)
30.000	0,05	600.000	1,08
200.000	0,36	800.000	1,46
400.000	0,73	1.000.000	1,80

Bảng 3.4 *Thời gian giải mã theo N, t và d=31*

N	*t*	*d*	Thời gian giải mã với *t* lấy theo *N* (giây)	Thời gian giải mã với *t* cố định (giây)	Độ chính xác
1.000	1.933	31	0,02	0,02	100%
3.000	2.240	31	0,05	0,08	100%
5.000	2.383	31	0,10	0,14	100%
7.000	2.477	31	0,12	0,16	100%
9.000	2.548	31	0,18	0,21	100%
11.000	2.604	31	0,23	0,26	100%
20.000	2.771	31	0,36	0,48	100%
40.000	2.965	31	0,71	1,01	100%
60.000	3.078	31	1,12	1,37	100%
80.000	3.159	31	1,51	1,84	100%
100.000	3.221	31	1,93	2,28	100%
120.000	3.272	31	2,36	2,79	100%
140.000	3.316	31	2,79	3,19	100%
160.000	3.353	31	3,22	3,65	100%
180.000	3.386	31	3,66	4,10	100%
200.000	3.415	31	4,11	4,56	100%
220.000	3.442	31	4,54	5,01	100%
240.000	3.466	31	5,00	5,48	100%
260.000	3.489	31	5,45	5,93	100%

Qua kết quả thực nghiệm ở các bảng 3.1, 3.2 và 3.3 cho thấy rằng trong các tham số của các mã RS sinh ma trận, để hỗ trợ số lượng IP lớn thì tham số *k* lớn. Khi giá trị *k* lớn thì *d* nhỏ, nghĩa là số lượng Hot-IP tối đa có thể phát hiện sẽ nhỏ.

Phần thử nghiệm kế tiếp sử dụng tham số *t* với chặn dưới của nó $t = \Omega(d^2 \frac{\log N}{\log d})$. Thông tin server chạy mô phỏng: CPU Intel Xeon E5-2650 2.00GHz. Giải pháp được cài đặt bằng ngôn ngữ lập trình C/C++ trên hệ điều hành CentOS 6.5, 64 bit. Từ các thời gian giải mã này có thể dùng làm cơ sở để cho lần lặp trong khoảng thời gian chu kỳ thuật toán để thực thi cho phù hợp với vị trí triển khai cụ thể nhằm giảm thời gian tính toán và tăng hiệu quả của giải pháp. Sinh ma trận d-phân-cách từ phép nối mã RS $[n_1, k]_q$ và I_q với n_1=63, k_1=3, q=64=2^6, t=n_1xq=4032, d=31, $N_{max}=q^k$=262144. Thử nghiệm thời gian giải mã với N=1.000, N=3.000, N=5.000, N=7.000, N=9.000, N=11.000, N=20.000, N=40.000, N=60.000, N=80.000, N=100.000, N=120.000, N=140.000, N=160.000, N=180.000, N=200.000, N=220.000 N=240.000, N=260.000. Giá trị *t* tương ứng được xác định bởi $t = \Omega(d^2 \frac{\log N}{\log d})$.

Kết quả thực nghiệm được trình bày trong bảng 3.4. Qua kết quả thực nghiệm cho thấy sự ảnh hưởng của kích thước ma trận đến thời gian giải mã, các ma trận có kích thước nhỏ cho kết quả thực hiện nhanh hơn các ma trận có kích thước lớn. Do đó, kích thước ma trận là vấn đề quan trọng cần được xem xét để lựa chọn ma trận phù hợp ở từng vị trí triển khai.

3.2.2. Lựa chọn các tham số

Trong giải pháp phát hiện các Hot-IP trên mạng sử dụng phương pháp thử nhóm bất ứng biến. Các tham số được lựa chọn như sau:

❖ **Xác định N**

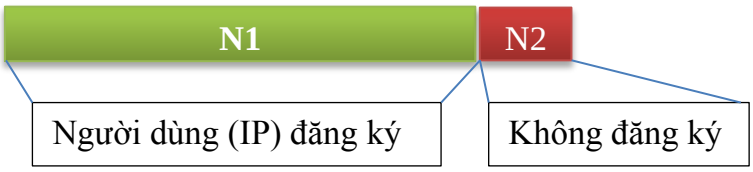
Kích thước ma trận trong thử nhóm bất ứng biến liên quan đến việc cập nhật dữ liệu đầu vào và các bước tính toán của giải pháp. Trong đó, giá trị N đại diện cho

số lượng địa chỉ IP phân biệt. Hai trường hợp áp dụng có thể xem xét để tính toán giá trị N được đề xuất như sau:

- **Trường hợp 1:** Xem xét các địa chỉ IP là như nhau

Trong trường hợp này, dựa vào khả năng của hệ thống tại vị trí triển khai và kinh nghiệm của người quản trị để xác định N trong một chu kỳ thuật toán.

- **Trường hợp 2:** Phân biệt các IP đăng ký và IP không đăng ký sử dụng dịch vụ



$$N = N1 + N2$$

Trong trường hợp này, có thể xem là một mạng cung cấp dịch vụ bên ngoài Internet. Số lượng người dùng đăng ký sử dụng dịch vụ là N1 và số lượng người dùng dịch vụ không đăng ký là N2. Đối với những người dùng không đăng ký có thể dùng với số lượng nhỏ bằng các địa chỉ đại diện.

Mục tiêu của việc phân chia này là nhằm (i) giới hạn giá trị N ở mức độ kiểm soát được và (ii) có thể ưu tiên sử dụng dịch vụ cho những người dùng đăng ký. Sự ưu tiên này thể hiện ở khả năng xuất hiện Hot-IP ở trong N2 rất lớn và khi đó có thể hạn chế truy cập đối với các địa chỉ này.

Vị trí đặt các bộ dò Hot-IP: trong giải pháp phát hiện các Hot-IP, bộ dò Hot-IP đặt trước đầu vào các router biên mạng (router gateway) đối với các hệ thống cung cấp dịch vụ trên Internet hoặc tích hợp vào các router trung gian ở các nhà cung cấp dịch vụ. Yêu cầu về thông lượng đối với các bộ dò này phải lớn hơn thông lượng của đường truyền tại chỗ đặt.

Hơn nữa, khi bị tấn công từ chối dịch vụ hay phát tán sâu mạng dạng quét không gian địa chỉ, thì tổng số lượng gói tin *m* tăng lên rất lớn, tuy nhiên tham số N

không lớn. Do vậy, trong giải pháp phát hiện các Hot-IP trực tuyến chu kỳ thực hiện thuật toán sẽ được chọn lựa để giá trị N phù hợp.

Dựa vào vị trí triển khai để xác định tham số *N*. Mỗi đơn vị triển khai, mỗi khu vực hoặc mỗi phân đoạn mạng nhỏ sẽ có số lượng giới hạn các IP cần quản lý. Gọi *N1* là số lượng IP mà ISP quản lý, gọi *n'* là số lượng IP khác (có thể là IP theo các khu vực, châu lục hay quốc gia). Ta có $n' \ll N1$ và $N = N1 + n'$.

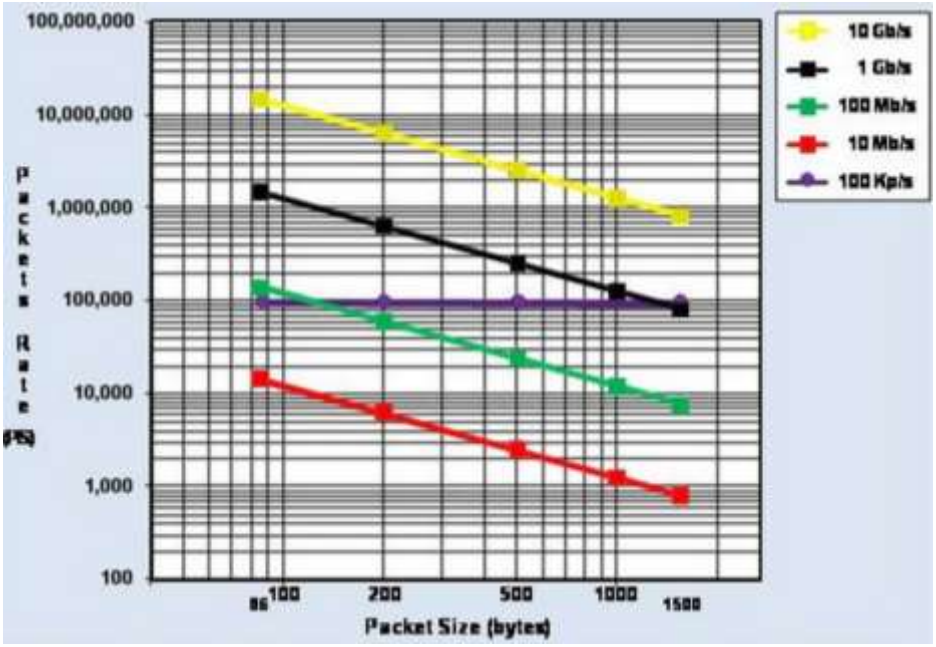
Phương pháp tóm tắt địa chỉ có thể được áp dụng để tối ưu không gian địa chỉ để giới hạn N. Từ các mạng hoặc mạng con (subnet) liên tục, chúng ta có thể tóm tắt chúng tạo thành địa chỉ mạng lớn hơn (super-network) làm địa chỉ đại diện. Bảng 3.5 mô tả ví dụ về phương pháp tóm tắt địa chỉ dùng làm địa chỉ đại diện nhằm giảm giá trị N.

Bảng 3.5 *Xác định địa chỉ đại diện cho các địa chỉ mạng*

Mạng	**Địa chỉ**	**Phương pháp**
Mạng 1	192.168.0.0/24	Xác định các bit chung (giống nhau) của các địa chỉ, cho các bit khác nhau bằng 0. Từ đó, xác định giá trị các octet và subnet-mask là số bit chung của các địa chỉ này.
Mạng 2	192.168.1.0/24	
Mạng 3	192.168.2.0/24	
Mạng 4	192.168.3.0/24	
Mạng 5	192.168.4.0/24	
Mạng 6	192.168.5.0/24	
Mạng 7	192.168.6.0/24	
Mạng 8	192.168.7.0/24	
Mạng đại diện	**192.168.0.0/21**	

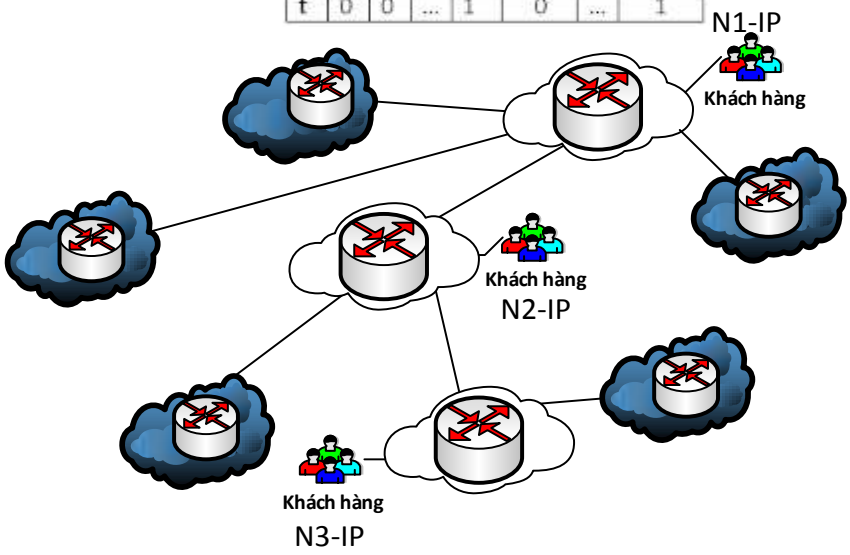
Mối liên hệ giữa tốc độ cổng giao tiếp (b/s) và số lượng gói tin truyền trong một đơn vị thời gian (p/s) cũng là yếu tố cần xem xét để xác định giới hạn của tham số N. Mối quan hệ này dựa trên việc xác định tốc độ cổng kết nối, tốc độ đường truyền, từ đó suy ra số lượng gói tin truyền theo thời gian theo kích thước tương

ứng cho từng loại kích thước gói tin. Trong mỗi gói tin (layer 3), chúng ta xác định được tương ứng thông tin địa chỉ IP. Đây cũng là cơ sở để xem xét trong việc lựa chọn kích thước ma trận phù hợp.



Hình 3.1. *Sự tương quan giữa bps và pps* [66]

	1	2	...	N1	N1+1	..	N1+N'
1	0	1	...	0	1	...	0
2	0	0	...	1	0	...	1
3	0	1	...	0	1	...	0
4	1	0	...	0	0	...	0
...	...		...	...	...	...	...
t	0	0	...	1	0	...	1



Hình 3.2. *Lựa chọn tham số N cho các bộ dò Hot-IP*

❖ **Xác định d**

Giá trị d trong ma trận d-phân-cách là số lượng Hot-IP tối đa có thể phát hiện được bằng phương pháp thử nhóm bất ứng biến. Tùy vào bài toán ứng dụng mà tham số này có ý nghĩa khác nhau. Trong tấn công từ chối dịch vụ phân tán, *d* có ý nghĩa là số lượng nguồn phát tấn công (giám sát dựa vào địa chỉ IP nguồn) hoặc là số lượng tối đa các server có khả năng bị tấn công (giám sát dựa vào địa chỉ IP đích). Trong ứng dụng phát hiện nguồn phát tán sâu Internet, giá trị *d* có ý nghĩa là số lượng tối đa các máy tính có khả năng đang quét mạng để phát tán sâu. Như vậy, *d* là giá trị được ước lượng định trước.

Từ các giá trị N và d được xác định, ma trận nhị phân d-phân-cách sinh ra từ phép nối mã được xác định dựa vào các công thức ràng buộc trong phương pháp thử nhóm như sau:

Với C_{out}: $[n_1,k]_q$-RS và I_q, ta có:

$$N_{\max} = q^k \tag{3.1}$$

$$t = n_1 \times q \tag{3.2}$$

$$d = \left\lfloor \frac{n_1 - 1}{k - 1} \right\rfloor \tag{3.3}$$

Với điều kiện $n_1 \le q$ và (3.2), (3.3) chọn giá trị n_1 và k để có giá trị d lớn. Trong các thực nghiệm của luận án sử dụng các ma trận d-phân-cách tường minh theo phương pháp nối mã dùng các C_{out} và C_{in}, với mã ngoài là mã Reed-Solomon C_{out}:[q-1, k] và mã trong là I_q.

Như vậy, bài toán thử nhóm bất ứng biến truyền thống phụ thuộc rất nhiều vào việc chọn *d*. Để giảm sự phụ thuộc vào *d*, luận án đề xuất phương án sử dụng danh sách Hot-List trong thuật toán cải tiến phương pháp thử nhóm bất ứng biến truyền thống được trình bày ở phần sau. Hot-List có kích thước lớn hơn *d*, là danh sách các địa chỉ IP nghi ngờ cùng với bộ đếm được khởi tạo tương ứng với nó. Thuật toán cải tiến đề xuất làm giảm thời gian tính toán và tăng độ chính xác trong

trường hợp số lượng Hot-IP trong dòng gói tin IP lớn hơn giá trị *d* đã xác định trước trong ma trận d-phân-cách. Kích thước của Hot-List là giá trị ước lượng cho số lượng Hot-IP tối đa giải pháp có thể phát hiện được.

❖ **Xác định m**

Tham số m_Δ là tổng số gói tin bắt được trong một chu kỳ thuật toán (Δ), là số lượng gói tin tối đa mà hệ thống (bộ phát hiện Hot-IP) có thể bắt được trong khoảng thời gian chu kỳ thuật toán.

Dựa vào mối tương quan giữa số lượng gói tin trên một giây (pps) và số lượng bit trên một giây (bps) đối với một số loại gói tin (packet size) có thể suy ra được số lượng gói tin tối đa trong một đơn vị thời gian (giây) tại các vị trí đặt thiết bị phát hiện Hot-IP. Mối tương quan giữa tốc độ cổng kết nối và số lượng gói tin bắt được trong một giây được thể hiện trên hình 3.1. Vị trí triển khai các bộ phát hiện Hot-IP ở vùng biên mạng, lấy giá trị nhỏ nhất giữa tốc độ cổng kết nối ra ngoài Internet và tốc độ đường truyền thuê bao Internet. Như vậy số lượng gói tin mà cổng kết nối vật lý có thể nhận được trong một chu kỳ thuật toán là:

m_Δ = (Số lượng gói tin tối đa cổng kết nối có thể nhận được/giây) x Δ

❖ **Ngưỡng tần suất xuất hiện cao** δ

Giá trị ngưỡng được đề cập ở đây là giá trị ngưỡng để xác định kết quả của một nhóm thử có chứa phần tử là Hot-IP hay không.

Ngưỡng tần suất suất cao được xác định dựa trên nghiên cứu của nhóm Cormode [27]:

$$\delta = \frac{m_\Delta}{(d+1)}$$

Luận án đề xuất sử dụng ngưỡng được tính như sau:

$$\delta = \frac{m_\Delta}{|\text{Hot-List}|}$$

Trong đó, m_Δ là tổng số gói IP bắt được trong thời gian tính toán. Ngưỡng được thiết lập như trên gọi là đặt ngưỡng theo khả năng của vị trí triển khai. Điều này thể hiện khả năng chịu đựng của hệ thống và phát huy được tối đa hiệu năng của hệ thống đối với các dòng dữ liệu lớn.

Ví dụ: Giả sử năng lực đáp ứng của hệ thống là 10^9 gói tin/giây và số Hot-IP dự đoán là 10^3. Khi đó ngưỡng sẽ là $\delta = \left\lfloor \frac{10^9}{10^3} \right\rfloor = 10^6$ với chu kỳ thuật toán $\Delta = 1$ giây.

❖ **Chu kỳ thuật toán (Δ):**

Trong bài toán phát hiện các Hot-IP trực tuyến, gọi thời gian thực hiện một chu kỳ thuật toán là Δ. Giá trị này có thể chọn 10 giây, 15 giây, 20 giây, 30 giây theo khả năng hệ thống, kinh nghiệm của người quản trị và có thể thay đổi bởi người quản trị.

❖ **Giá trị t**

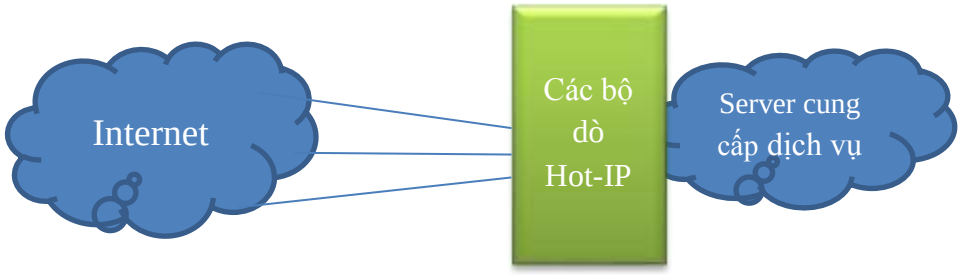
Giá trị *t* là số hàng của ma trận hay số nhóm thử được thiết kế trước theo phương pháp thử nhóm bất ứng biến. Trong xây dựng ma trận d-phân-cách bằng phương pháp nối mã, giá trị *t* được xác định như sau: $t = n_1 \times q,$ với $C_{out} : [n_1, k_1]_q$-RS và $C_{in} : I_q$.

3.3. KIẾN TRÚC PHÂN TÁN

3.3.1. Giới thiệu

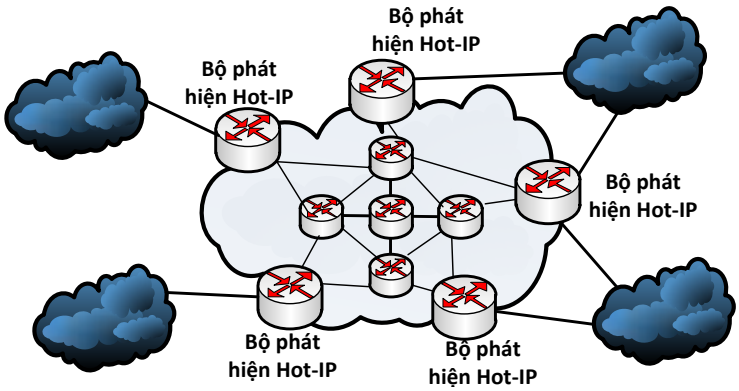
Bảo vệ các ứng dụng, dịch vụ trên Internet trước các cuộc tấn công từ chối dịch vụ là vấn đề quan trọng trong bài toán an ninh mạng. Các cuộc tấn công mạng ngày càng có tính phối hợp cao, phân tán rộng trên Internet, để phát hiện và phòng chống một cách hiệu quả thì chiến lược phát hiện và phòng chống cũng cần được triển khai phân tán và hợp tác giữa các thành phần. Điều này đòi hỏi việc thu thập thông tin về tấn công một cách rộng rãi để có thể phát hiện chính xác, kịp thời.

Sử dụng các bộ phát hiện Hot-IP (hay bộ dò Hot-IP) làm vai trò gateway đứng trước hệ thống máy chủ cung cấp ứng dụng, dịch vụ trên mạng có ý nghĩa quan trọng: (i) cách ly giữa tấn công và ứng dụng, ngăn chặn trực tiếp những cuộc tấn công DoS/DDoS ở mức cơ sở hạ tầng lên ứng dụng và (ii) dùng nhiều bộ phát hiện Hot-IP để phân tán lưu lượng, làm giảm ảnh hưởng của cuộc tấn công.



Hình 3.3. *Vị trí đặt bộ dò ở gateway của hệ thống mạng*

Trên mạng tốc độ cao, để giảm tải dòng dữ liệu trên các thiết bị định tuyến cần phải sử dụng các thiết bị chia tải trên luồng lưu lượng. Với dòng dữ liệu thời gian thực, việc chia tải này sẽ giúp hệ thống đáp ứng tốt hơn, tránh tắc nghẽn và giảm độ trễ trên đường truyền.



Hình 3.4. *Kiến trúc phân tán các ngõ vào của hệ thống*

Trong các hệ thống mạng đa vùng, các bộ phát hiện Hot-IP có thể triển khai ở khu vực biên mạng ở mỗi vùng. Điều này cho phép hệ thống chịu đựng được các tấn công DoS/DDoS, hay các cuộc quét mạng quy mô lớn của một số loại sâu để tìm kiếm lỗ hổng của các thiết bị trên Internet bằng lưu lượng tấn công phân tán. Bên cạnh khả năng bảo vệ tính sẵn sàng của ứng dụng, giải pháp triển khai dạng

phân tán còn có khả phòng thủ DoS/ DDoS, phù hợp triển khai ở phía các mạng trung gian ISP hay các nhà cung cấp dịch vụ trên Internet.

Hình 3.4 mô tả hệ thống mạng các bộ phát hiện Hot-IP trung gian đối với tất cả lưu lượng giữa ứng dụng và người sử dụng, bảo vệ ứng dụng trước các cuộc tấn công từ chối dịch vụ. Trong đó, thành phần hệ thống quan trọng gồm ứng dụng, người dùng, máy chủ và mạng bộ phát hiện Hot-IP.

Kiến trúc phân tán có thể áp dụng cho bài toán phát hiện các Hot-IP nhằm giảm tải trong việc xử lý và tính toán trong chương trình. Các thiết bị định tuyến đặt ở khu vực biên mạng thu thập dòng dữ liệu đầu vào và gửi kết quả gồm tên đối tượng cùng với tần suất tương ứng cho thiết bị định tuyến trung tâm. Thiết bị định tuyến trung tâm tập hợp dữ liệu phân tán và đưa vào ma trận *d*-phân-cách, từ đó tính toán để dò tìm các phần tử Hot-IP trên mạng.

Ở các nhà cung cấp dịch vụ (ISP) hoặc các đơn vị cung cấp dịch vụ ứng dụng ngoài Internet trên phạm vi rộng lớn, hệ thống mạng được tổ chức thành các khu vực. Mỗi khu vực phục vụ cho các khách hàng trong khu vực đó. Các bộ phát hiện Hot-IP được thiết lập cho từng vị trí triển khai để phát hiện các Hot-IP trên cơ sở giới hạn các địa chỉ được giám sát (các địa chỉ IP của khách hàng mà vị trí triển khai quản lý). Thiết lập kiến trúc phân tán cho bài toán phát hiện nhanh Hot-IP là một giải pháp hiệu quả nhằm nâng cao khả năng tính toán và phù hợp với kiến trúc phân tán của các hệ thống mạng được tổ chức đa khu vực như các ISP.

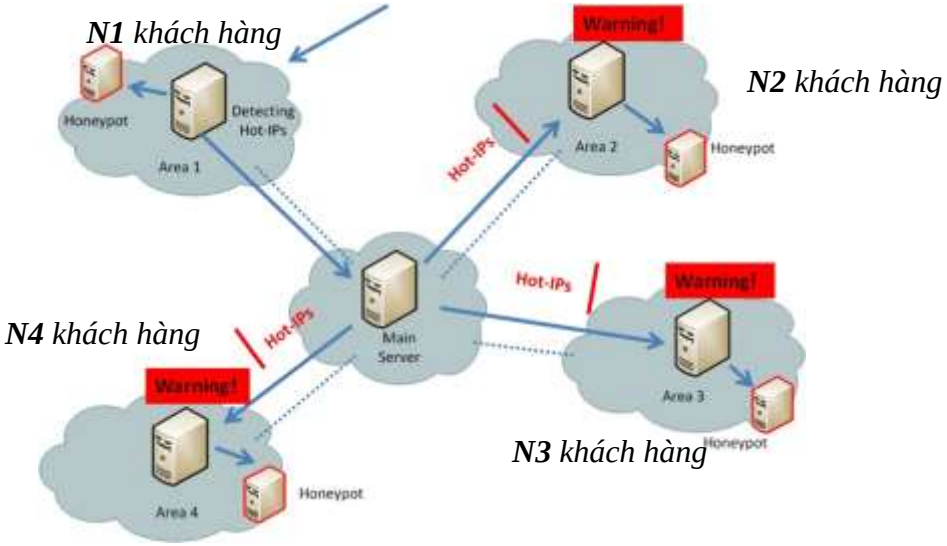
Việc thiết lập kiến trúc phân tán trong giải pháp dò tìm các đối tượng có khả năng là nguồn phát tán sâu mạng hay tấn công từ chối dịch vụ đã được đề cập trong nhiều nghiên cứu. Đây thực sự là một giải pháp kỹ thuật hiệu quả để phát hiện sớm các nguy hại trên hệ thống mạng. Việc phát hiện các nguy hại này ở mỗi vị trí triển khai sẽ giúp nhanh chóng chuyển tiếp các cảnh báo đến các khu vực khác. Từ đó giúp người quản trị nhanh chóng có giải pháp ứng phó kịp thời.

3.3.2. Kiến trúc phân tán phát hiện Hot-IP

Kiến trúc phân tán trong giải pháp phát hiện các Hot-IP có ý nghĩa quan trọng và phù hợp với kiến trúc mạng của các hệ thống tổ chức theo dạng đa khu vực hiện nay của các doanh nghiệp hoặc mạng của các nhà cung cấp dịch vụ Internet.

Hệ thống mạng của ISP được tổ chức thành các khu vực, ở mỗi khu vực cung cấp dịch vụ cho các khách hàng trong khu vực đó. Ở mỗi khu vực, hệ thống các bộ phát hiện Hot-IP được thiết lập để phát hiện các Hot-IP (gọi là vị trí triển khai giải pháp). Hệ thống bộ phát hiện Hot-IP ở các vị trí triển khai giải pháp được thiết kế để kết nối với nhau theo dạng ngoài luồng dữ liệu hoặc tích hợp vào các router biên mạng. Mục đích của việc thiết lập này để tăng khả năng phát hiện sớm các Hot-IP trong mạng và trách tắc nghẽn khi có tấn công xảy ra.

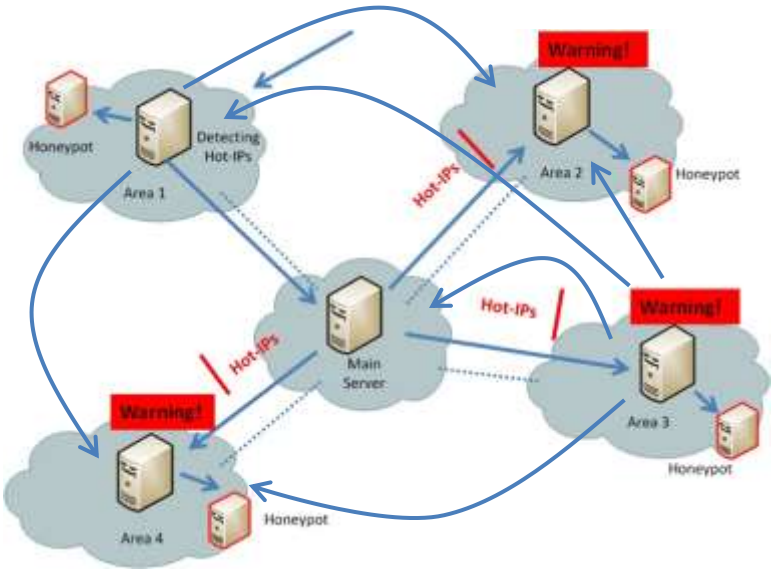
Giả sử ở vị trí triển khai giải pháp tại *Area 1* quản lý N1 khách hàng và n1' địa chỉ IP cần giám sát (địa chỉ đại diện cho một số khu vực, quốc gia,…). Như vậy, số lượng địa chỉ IP cần thiết để quan sát trong giải pháp ở vị trí *Area 1* là (N1+n1'). Tương tự cho các khu vực khác, lựa kích thước ma trận phù hợp với khả năng xử lý của thiết bị triển khai giải pháp và tùy vào lượng khách hàng mà khu vực đó quản lý. Sự kết nối giữa các bộ phát hiện Hot-IP có thể được triển khai quản lý tập trung như trong hình 3.5 hay theo dạng ngang hàng như trong hình 3.6.



Hình 3.5. *Kiến trúc phân tán với các bộ phát hiện Hot-IP được quản lý tập trung*

Trong mô hình các bộ phát hiện Hot-IP được quản lý tập trung, bộ phát hiện Hot-IP trung tâm được đặt ở khu vực trung tâm, các bộ phát hiện Hot-IP ở từng khu vực đóng vai trò như bộ cảm biến, định thời kiểm tra để phát hiện các Hot-IP trên mạng. Nếu phát hiện ra Hot-IP sẽ gửi cảnh báo đến bộ phát hiện Hot-IP trung tâm và các bộ phát hiện Hot-IP ở các khu vực khác tùy vào mục đích của việc giám sát. Bộ phát hiện Hot-IP trung tâm cũng hoạt động như một bộ cảm biến để phát hiện các Hot-IP, đồng thời là điểm tập trung quản lý các bộ phát hiện Hot-IP ở các khu vực. Kết nối giữa các bộ phát hiện Hot-IP này thực hiện theo dạng ngoài luồng dữ liệu, nghĩa là đi theo đường kết nối riêng nhằm mục đích tránh tắc nghẽn khi đi chung với đường truyền dữ liệu và tăng tốc trong quá trình phát cảnh báo.

Để tránh bộ phát hiện Hot-IP ở vị trí trung tâm có thể bị tấn công, kiến trúc giao tiếp ngang hàng của các bộ phát hiện Hot-IP ở các khu vực có thể thông tin trực tiếp với nhau sẽ giúp cho hệ thống giải pháp hoạt động hiệu quả hơn. Trong đó, bộ phát hiện Hot-IP ở mỗi khu vực sẽ tạo kết nối đến một số bộ phát hiện Hot-IP ở các khu vực khác. Khi một bộ phát hiện Hot-IP phát hiện có Hot-IP sẽ phát cảnh báo cho các bộ phát hiện Hot-IP khác có thiết lập kế nối với nó. Hình 3.6 mô tả kiến trúc phân tán và giao tiếp ngang hàng được thiết lập giữa các bộ dò Hot-IP.



Hình 3.6. *Kiến trúc phân tán và giao tiếp ngang hàng giữa các bộ dò Hot-IP*

Như vậy, trong mô hình các bộ phát hiện Hot-IP được tập trung quản lý tại trung tâm giúp quá trình kiểm soát và điều khiển các bộ phát hiện Hot-IP tốt hơn, tuy nhiên hệ thống bộ phát hiện Hot-IP có thể sẽ mất kiểm soát trong việc điều phối hoạt động một khi bộ phát hiện Hot-IP trung tâm bị tấn công. Trong mô hình các bộ phát hiện Hot-IP hoạt động theo cơ chế ngang hàng sẽ linh động hơn, tránh hạn chế như trong mô hình tập trung.

3.3.3. Kịch bản thực nghiệm và kết quả

Trong phần thực nghiệm, luận án sử dụng 4 server gồm 1 server đóng vai trò là bộ phát hiện Hot-IP trung tâm và 3 server còn lại đóng vai trò là các bộ phát hiện Hot-IP thành viên đặt ở các khu vực khác nhau, sử dụng lập trình mạng dạng client/server để tạo sự kết nối giữa các server này nhằm mục đích sẽ thông báo cho nhau khi một server phát hiện có Hot-IP trên mạng.

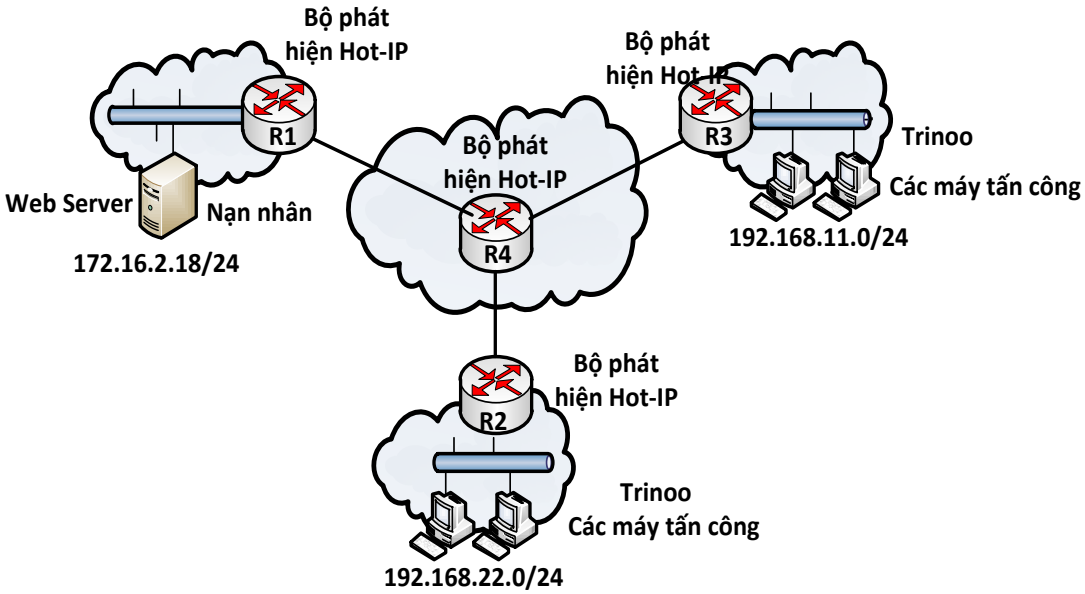
Dòng gói tin IP được thu thập vào các server, địa chỉ IP nguồn và IP đích được trích ra từ IP-header ở mỗi gói tin để xử lý. Ở mỗi vị trí triển khai sử dụng ma trận phù hợp với lượng IP quản lý cộng với một số lượng nhỏ IP đại diện cho các đối tượng khác như các ISP khác, các quốc gia hay khu vực.

Để đánh giá khả năng của thuật toán cải tiến đề xuất, trong phần thực nghiệm luận án sử dụng các ma trận ở các vị trí triển khai hỗ trợ lượng IP khác nhau: 4096, 32768, 262144, 33554432. Ma trận phân cách được sinh ra bằng phương pháp nối mã tương ứng sử dụng từ các bộ mã Reed-Solomon $[7,3]_8$-RS (d=7, N=4.096, t=240), $[31,3]_{32}$-RS (d=15, N=32.768, t=992), $[63,3]_{64}$-RS và $[31,5]_{32}$-RS (d=7, N=33554432, t=992). Chu kỳ thực hiện thuật toán $\Delta = 10$ giây, $\Delta = 15$ giây, $\Delta = 20$ giây, $\Delta = 25$ giây, $\Delta = 30$ giây. Thuật toán sinh ra các gói tin chứa địa chỉ IP của người những dùng bình thường và các gói tin có tần suất cao được phát sinh từ các phần mềm tấn công như Trinoo.

Ở mỗi khu vực, bộ phát hiện Hot-IP định thời kiểm tra trên dòng dữ liệu để phát hiện các Hot-IP. Khi phát hiện có Hot-IP, bộ phát hiện sẽ phát cảnh báo đến các khu vực khác. Trong phần thực nghiệm này, liên kết được thiết lập giữa các bộ

phát hiện Hot-IP ở các khu vực để trao đổi kết quả phát hiện Hot-IP giúp các khu vực nhận được kết quả mà không cần phải thực hiện tính toán.

Trong phần thực nghiệm này, sử dụng 10 máy chạy chương trình quét cổng, sử dụng mạng IPv4 và không gian địa chỉ 2^{32}. Phương pháp quét được sử dụng gồm có quét TCP, quét UDP, kết quả trả về là thông tin hệ thống chạy trên các máy nạn nhân phát hiện được.



Hình 3.7. *Sơ đồ thực nghiệm kiến trúc phân tán phát hiện các Hot-IP*

Các tham số sử dụng trong thực nghiệm với dữ liệu thực tế: chu kỳ thuật toán $\Delta = 30$ giây, số lượng IP phân biệt trong hệ thống giám sát là N = 4096, ma trận phân cách M có kích thước 240x4096, số lượng Hot-IP có hệ thống có khả năng phát hiện ước lượng là 100, các cổng giao tiếp của các kết nối là 100Mbps. Dựa trên tốc độ của các kết nối, suy ra được $m_\Delta = 50.000 \times 30 = 1.500.000$ và ngưỡng tần suất cao $\delta_\Delta = \dfrac{m_\Delta}{|\text{Hot-List}|} = 15.000.$

Các trường hợp thực nghiệm như sau:

Trường hợp 1: Chương trình được cài đặt tại các vị trí R1, R2, R3, R4 có khả năng phát hiện và phát cảnh báo cho nhau khi phát hiện có Hot-IP. Thuật toán cải tiến 2 "Online Hot-IP Preventing" được sử dụng để tại mỗi vị trí triển khai khi phát

hiện Hot-IP sẽ ngăn chặn các gói tin chứa IP này trong một chu kỳ thuật toán. Đồng thời, địa chỉ Hot-IP sẽ được gửi tới bộ dò của khu vực chứa IP nạn nhân và thực hiện ngăn chặn các IP này trong một chu kỳ thuật toán. Trong trường hợp này, các bộ dò R2 và R3 phát hiện các Hot-IP và khóa các IP này trong một chu kỳ thuật toán. Lưu lượng ở R1 và R4 bình thường, các bộ phát hiện R4 và R1 không phát hiện được các Hot-IP.

Trường hợp 2: Các bộ phát hiện Hot-IP được triển khai thực nghiệm trên R1 và R4, đại diện cho khu vực mạng trung gian (R4) và khu vực chứa nạn nhân (R1). Tấn công xuất phát từ 2 khu vực R2 và R3, bộ phát hiện ở R4 phát hiện được nạn nhân là IP của Web server và cảnh báo cho R1 các Hot-IP phát hiện được, đồng thời ngăn chặn các Hot-Ip này trong một chu kỳ thuật toán. Trong trường hợp triển khai ở R4, việc phát hiện và ngăn chặn có thể làm tăng mức độ xử lý trên R4, nếu chỉ phát hiện và gửi thông tin của Hot-IP đến bộ dò gắn với phía nạn nhân R1 ngăn chặn thì giảm được mức độ xử lý đối với các router trung gian

Trường hợp 3. Các bộ dò Hot-IP được triển khai ở các khu vực mà không triển khai ở các router trung gian. Trong trường hợp này, các bộ phát hiện được cài đặt trên R1, R2 và R3. Trường hợp này phù hợp cho các hệ thống của công ty triển khai các dịch vụ cung cấp cho người dùng trên Internet, đặt các server ở nhiều khu vực để cung cấp dịch vụ cho người dùng. Khi bộ phát hiện Hot-IP phát hiện có Hot-IP sẽ tiến hành ngăn chặn và phát cảnh báo cho các bộ phát hiện khác trong hệ thống để tiến hành ngăn chặn trong một chu kỳ thuật toán.

Trong các trường hợp trên, giải pháp phát hiện các Hot-IP có thể kết hợp thêm một số chức năng khác để hệ thống linh hoạt hơn như chuyển luồng lưu lượng sang các server cung cấp ở các khu vực khác hay chuyển các Hot-IP vào các Honeypot để phòng chống tấn công. Các thực nghiệm cài đặt thuật toán cải tiến 2 “Online Hot-IP Preventing” cho thấy giải pháp có khả năng ứng dụng nhằm ngăn ngừa sớm các nguy cơ có thể xảy ra như các tấn công DoS/DDoS, giúp hệ thống hoạt động ổn định, thông suốt.

3.4. GIẢI PHÁP SONG SONG

3.4.1. Giới thiệu

Tính toán song song hay xử lý song song là quá trình xử lý gồm nhiều tiến trình được kích hoạt đồng thời và cùng tham gia giải quyết một vấn đề. Xử lý song song được sử dụng để tăng tốc độ tính toán trong việc giải quyết các bài toán lớn, phức tạp để nhanh chóng cho ra kết quả.

Kiến trúc song song là một giải pháp hữu hiệu để tăng năng lực cho các hệ thống tính toán, trong đó nhiều đơn vị dữ liệu được xử lý đồng thời bởi một hay nhiều bộ xử lý để giải quyết một bài toán. Lĩnh vực xử lý song song là một lĩnh vực thú vị, được nhiều nhà khoa học quan tâm để giải quyết các bài toán có khối lượng tính toán lớn. Mục đích chính của giải pháp này là nâng cao năng lực xử lý, tính toán cho một bài toán phức tạp để đạt kết quả trong thời gian nhanh hơn so với việc xử lý theo một thuật toán thông thường. Phương pháp này được áp dụng bằng cách phân chia tác vụ lớn thành những tác vụ nhỏ, phân bổ tính toán cho các máy khác trong hệ thống và tổng hợp kết quả trả về.

Một trong những phân loại hay được nhắc tới là của Flynn – 1972. Michael Flynn phân các kiến trúc máy tính thành bốn loại dựa trên tương tác giữa lệnh và dữ liệu :

- **SISD** (single instruction stream, single data stream): là kiến trúc tuần tự Von Neuman, trong đó tại mỗi thời điểm chỉ một lệnh được thực hiện.
- **MISD** (multiple instruction stream, single data stream): Kiến trúc này cho phép nhiều lệnh cùng thao tác trên một dữ liệu.
- **SIMD** (single instruction stream, multiple data stream): Cho phép một lệnh được thực hiện đồng thời trên các dữ liệu khác nhau.
- **MIMD** (multiple instruction stream, multiple data stream): Cho phép nhiều lệnh khác nhau có thể đồng thời xử lý nhiều dữ liệu khác nhau trong cùng một thời điểm.

Các thuật toán song song được thực hiện trên một tập các bộ xử lý nên đòi hỏi việc truyền dữ liệu giữa chúng. Vì thế, nó bao hàm hai hoạt động khác nhau, một hoạt động là tính toán được thực hiện một cách cục bộ trên một bộ xử lý, hoạt động nữa là gửi dữ liệu giữa các bộ xử lý.

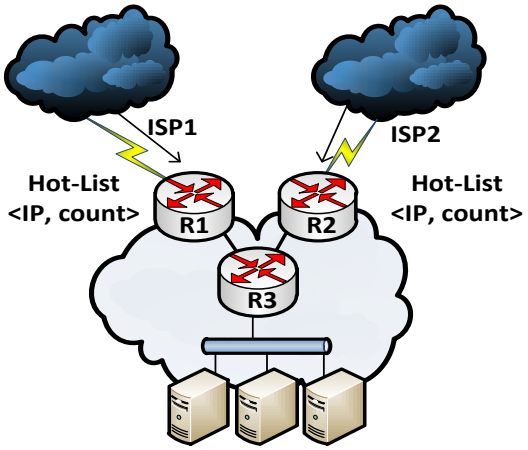
Trong các ứng dụng thời gian thực, việc xác định nhanh các đối tượng mục tiêu là vô cùng quan trọng. Các cuộc tấn công từ chối dịch vụ có quy mô ngày càng lớn và gây hậu quả nghiêm trọng, cũng như các cuộc quét mạng tìm kiếm lỗ hổng để phát tán sâu Internet diễn ra với tốc độ rất nhanh, thời gian thực hiện quét mạng ngắn. Mô hình hóa các đối tượng này về bài toán phát hiện các Hot-IP có nhiều ý nghĩa to lớn. Từ đó, việc xác định nhanh các Hot-IP là cơ sở giúp nhanh chóng xác định các nguồn phát tấn công, các nạn nhân trong cuộc tấn công này, các máy nhiễm sâu đang tiến hành quét mạng để lây lan, giúp người quản trị mạng có thời gian để đưa ra các giải pháp hợp lý, kịp thời. Áp dụng kỹ thuật xử lý song song để tăng tốc trong bước giải mã xác định các Hot-IP là một trong những giải pháp hữu hiệu để nâng cao khả năng áp dụng của giải pháp vào thực tế.

3.4.2. Xử lý song song trong bài toán thử nhóm

Bài toán xác định các Hot-IP trên mạng bằng phương pháp thử nhóm bất ứng biến có khối lượng tính toán lớn, mất nhiều thời gian xử lý, phụ thuộc vào kích thước của ma trận *d*-phân-cách. Trong tổng thời gian thực hiện chương trình, việc xử lý các gói tin đầu vào, trích thông tin địa chỉ IP và việc làm giảm thời gian giải mã có ý nghĩa hết sức quan trọng trong dòng gói tin IP thời gian thực để có thể phát hiện nhanh và tiến hành các giải pháp hạn chế rủi ro có hiệu quả.

❖ **Xử lý ở bước thu thập dữ liệu đầu vào:**

Giải pháp thu thập dữ liệu từ các thiết bị có thể tích hợp vào các ngõ vào của hệ thống như các router biên được thiết kế theo dạng chia tải cho toàn hệ thống.

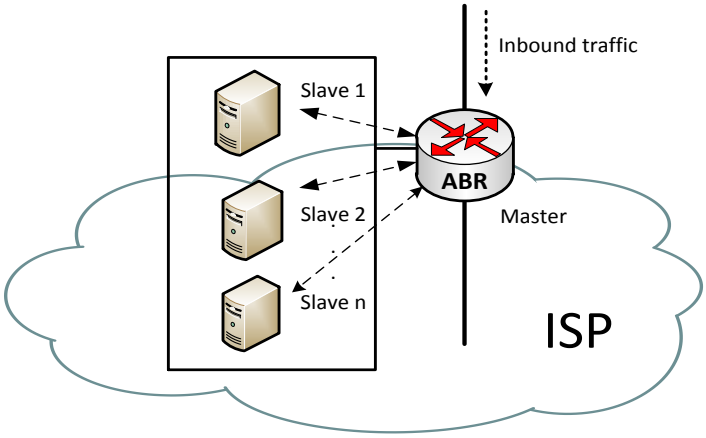


Hình 3.8. *Thu thập dữ liệu đầu vào dạng phân tán*

Các router biên trong hệ thống nhận dữ liệu vào (R1 và R2) được thiết bị R3 xử lý và điều phối hoạt động cập nhật địa chỉ trong các gói tin vào chương trình. Giải pháp phân tán xử lý các dữ liệu đầu vào thời gian thực được sử dụng để giải quyết bài toán xử lý với luồng dữ liệu lớn để tăng thời gian đáp ứng của hệ thống cho các yêu cầu truy xuất bên ngoài hệ thống.

❖ **Xử lý ở bước tính vector kết quả cho các nhóm thử:**

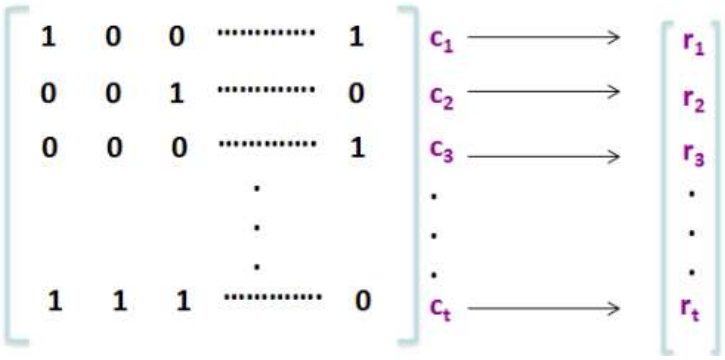
Các router biên mạng tiếp nhận và xử lý các gói tin đi qua nó, để giảm tải các xử lý trên các router này trong việc thực hiện giải pháp phát hiện các Hot-IP, có thể thiết lập tính toán song song bằng cách phân chia tác vụ cho các thiết bị tính toán bên trong nó. Từ đó giảm tải tính toán và giảm thời gian tính toán cho toàn bộ giải pháp.



Hình 3.9. *Mô hình tính toán song song kết nối giữa router biên và các server*

Việc mô phỏng giải pháp tính toán song song được tác giả thử nghiệm và kết quả cho thấy đây là giải pháp có thể ứng dụng để giảm thời gian tính toán. Theo mô hình thử nghiệm, để tiến hành các xử lý này cần có các thiết bị liên quan phối hợp xử lý. Như vậy, để triển khai giải pháp tính toán song song, việc xác định các thiết bị phối hợp này cũng cần tính toán hợp lý.

Từ bài toán tìm các Hot-IP bằng phương pháp thử nhóm bất ứng biến cho thấy rằng việc tính tổng số các gói tin từng nhóm, so sánh số gói tin đó với giá trị ngưỡng để xác định kết quả của phép thử được lặp đi lặp lại nhiều lần và các nhóm thử được thiết kế độc lập nhau. Như vậy, ở bước xác định các kết quả của phép thử có thể sử dụng kỹ thuật xử lý song song để tối ưu thời gian tính toán kết quả của các nhóm thử.



Hình 3.10. *Song song các bước tính toán kết quả các nhóm thử*

Thuật toán xử lý song song:

- *Gọi: N* là tổng số IP phân biệt trong dòng gói tin IP trong khoảng thời gian delta(t), *m* là tổng số gói tin trong dòng gói tin IP, M_{txN} là ma trận d-phân-cách.
- *Máy Master:* khởi tạo
  - M_{txN} //ma trận d-phân-cách
  - *idle=0* //khởi tạo dòng hiện hành đang giao xử lý
  - *ntasks=t* //số hàng của ma trận
- *Ở các bộ xử lý song song: Tính vector kết quả của nhóm thử R(i) ở các bộ xử lý*

```
1:  for each processor i, in parallel
2:       do if not (ntasks=0))
3:            then
4:                 if C(i)> δ then
5:                      R(i)=1
6:                 else
7:                      R(i)=0
8:                      ntasks=ntasks-1
```

<Các R(i) được gửi về cho Master tổng hợp và xác định các Hot-IP>

Phần thực nghiệm áp dụng mô hình xử lý song song, luận án cài đặt theo mô hình master/slave của PVM để song song hóa các bước tìm vector kết quả như sau:

- Số tiến trình tương ứng với số máy *Slave* tham gia hệ thống
- Máy *Master* có nhiệm vụ gửi dữ liệu cho các máy *slave* để tính toán
- Số tác vụ cần phải thực hiện tương ứng với số hàng của ma trận
- Mỗi *Slave* sẽ phải tính toán để tìm ra kết quả phép thử và trả kết quả đó về cho *Master*
- *Master* sẽ tổng hợp kết quả trả về từ các *Slave* và xác định các Hot-IP

3.4.3. Kịch bản thực nghiệm và kết quả

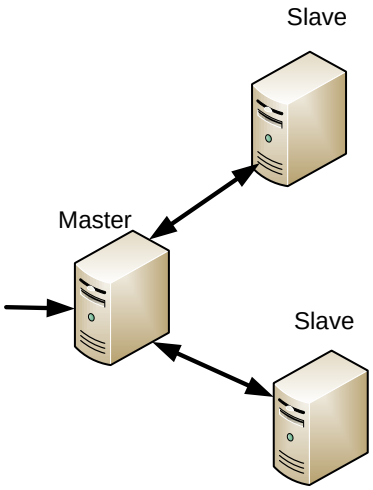
❖ **Thực nghiệm xử lý song song dữ liệu đầu vào**

Để xử lý nhanh các luồng dữ liệu rất lớn đối với việc xử lý dữ liệu đầu vào, phần thực nghiệm sử dụng công cụ MapReduce (Hadoop) trên các router R1, R2, để thu thập thông tin IP trên các dữ liệu đầu vào. R3 hoạt động như thiết bị chia tải, nhận và xử lý luồng dữ liệu tổng hợp, thuật toán cải tiến 2 "Online Hot-IP Preventing" được sử dụng để phát hiện và ngăn chặn các Hot-IP.

❖ **Thực nghiệm xử lý song song ở bước tính toán kết quả**

Trong mô hình thực nghiệm xử lý song song, luận án sử dụng 1 máy làm chức năng *Master* và 2 máy làm *Slave* với các thông số cấu hình như sau:

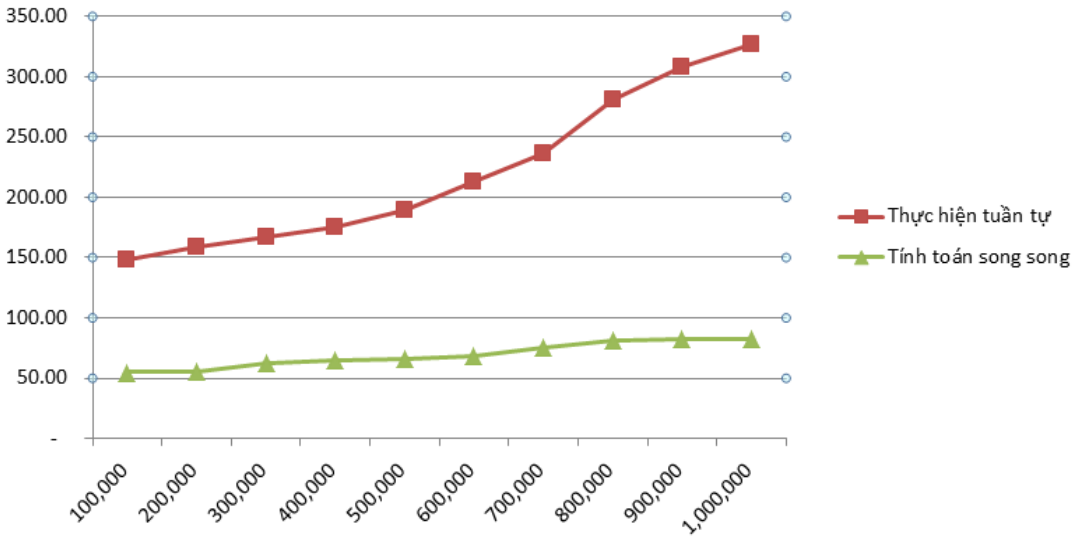
Master	2 máy Slave
- Core i5-2410 CPU 2.3 GHz - Bộ nhớ: 1GB - Hệ điều hành: CentOS	- Intel Pentium 4 CPU 2.4 GHz - Bộ nhớ: 256 MB - Hệ điều hành: CentOS



Hình 3.11. *Mô hình thực nghiệm xử lý song song*

❖ **Kịch bản:** ma trận d-phân-cách được sinh ra từ mã RS $[15,5]_{16}$ và ma trận đơn vị I_{16} với d=3 ($d=3, N=1.048.576,\ t=240$). Thiết lập các cuộc tấn công từ chối dịch vụ với số lượng IP (100.000 →1.000.000) và tính toán thời gian giải mã trong trường hợp xử lý từng tự và xử lý song song để so sánh.

❖ **Kết quả thực nghiệm:**



Hình 3.12. *Biểu đồ thời gian giải mã xác định các Hot-IP*

Bảng 3.6. *Kết quả thực nghiệm xử lý tuần tự và song song*

N	Thực hiện tuần tự (giây)	Tính toán song song (giây)	Chênh lệch (giây)
100.000	148,02	54,73	93,29
200.000	159,15	55,07	104,08
300.000	166,91	61,84	105,07
400.000	175,69	64,95	110,74
500.000	189,83	65,48	124,35
600.000	212,76	68,74	144,02
700.000	236,36	75,33	161,03
800.000	281,10	80,97	200,13
900.000	308,46	82,41	226,05
1.000.000	327,12	82,71	244,41

Kết quả thực nghiệm được thể hiện trên hình 3.12 và bảng 3.6 cho thấy phương pháp xử lý song song cho kết quả giải mã nhanh hơn nhiều so với xử lý tuần tự. Từ đó cho thấy rằng với việc xây dựng giải pháp phát hiện nhanh các Hot-IP trên mạng dùng phương pháp thử nhóm bất ứng biến kết hợp với kỹ thuật xử lý song song cho kết quả rất tốt, có khả năng áp dụng hiệu quả trong triển khai thực tế trên các mạng tốc độ cao.

Như vậy, hệ thống tính toán song song được đề xuất áp dụng trong giải pháp phát hiện nhanh các Hot-IP trên mạng đã được cấu hình thử nghiệm cho thấy mức độ hiệu quả trong việc giảm thời gian phát hiện các Hot-IP và mới chỉ dừng ở mức độ mô phỏng. Để có thể triển khai được vào thực tiễn cần có những phân tích và thực nghiệm kỹ hơn với những dữ liệu thực tế, vị trí triển khai thực tế để xác định các tham số cũng như số lượng các bộ xử lý phù hợp nhằm đạt hiệu quả cao khi áp dụng tính toán song song trong giải pháp phát hiện các Hot-IP trực tuyến trên mạng.

3.5. KẾT LUẬN CHƯƠNG 3

Trong chương này, luận án trình bày một số kỹ thuật kết hợp để nâng cao hiệu quả của giải pháp phát hiện các Hot-IP trên mạng. Trong đó, luận án xác định phương pháp lựa chọn các tham số cho phương pháp thử nhóm bất ứng biến t, d, N. Mục tiêu của việc lựa chọn này là lựa chọn kích thước ma trận hợp lý tùy thuộc vào vị trí triển khai.

Các nhóm thử trong thử nhóm bất ứng biến tương ứng với các dòng của ma trận nhị phân d-phân-cách, các phép thử này là độc lập nhau. Do đó, việc tính toán kết quả cho mỗi phép thử có thể áp dụng kỹ thuật xử lý song song để nâng cao hiệu quả tính toán trong chương trình. Với việc tổ chức hệ thống mạng theo dạng đa vùng của các nhà cung cấp dịch vụ và các ứng dụng phổ biến trên Internet, giải pháp triển khai phân tán các bộ phát hiện Hot-IP ở mỗi khu vực và thiết lập chế độ giao tiếp giữa chúng nhằm cảnh báo sớm các nguy cơ tấn công hoặc mục tiêu tấn công có ý nghĩa quan trọng trong bài toán an ninh mạng. Các nội dung chính của chương trình được công bố trong các công trình [C4][C6][C7] trong danh mục các công trình nghiên cứu của tác giả.

Tấn công từ chối dịch vụ và sâu Internet là mối đe dọa lớn đến an ninh mạng toàn cầu, chúng không thể được giải quyết thông qua những hành động tự lập của những nút mạng phòng chống tấn công triển khai rải rác. Bài toán phát hiện các Hot-IP là bài toán tổng quát hóa của các mối đe dọa nói trên. Có thể xem giải pháp phát hiện các Hot-IP là giải pháp phòng ngừa giúp làm giảm số lượng máy có nguy cơ bị tấn công nên hạn chế được sự lây lan. Những hệ thống phòng thủ phải được tổ chức vào một mô hình liên kết động để đảm bảo hệ thống hoạt động ổn định, thông suốt. Trong chương 4 sẽ trình bày chi tiết hơn về một số ứng dụng từ bài toán phát hiện các Hot-IP trên mạng.

CHƯƠNG 4. MỘT SỐ ỨNG DỤNG PHÁT HIỆN CÁC HOT-IP

4.1. GIỚI THIỆU

Bài toán phát hiện các Hot-IP trực tuyến là bài toán tổng quát, có thể ứng dụng vào một số bài toán an ninh mạng. Xác định các Hot-IP chính là xác định các đối tượng trên mạng hoạt động với tần suất cao trong một khoảng thời gian rất ngắn. Các đối tượng này có khả năng là nguy cơ ảnh hưởng đến hoạt động của hệ thống mạng, có thể là nguồn phát hay mục tiêu trong các cuộc tấn công từ chối dịch vụ, có thể là các máy tính đang quét mạng để tìm kiếm lỗ hổng nhằm phát tán sâu mạng của một số loại sâu Internet hoạt động theo dạng quét không gian địa chỉ IP.

Ở chương 3 đã trình bày một số kỹ thuật kết hợp để nâng cao hiệu quả của giải pháp phát hiện các Hot-IP trên mạng. Những kỹ thuật kết hợp được đề cập trong chương 3 nên được xem xét kỹ khi áp dụng ở các vị trí triển khai cụ thể để đạt hiệu quả cao. Chương này sẽ trình bày một số ứng dụng của bài toán phát hiện các Hot-IP trực tuyến trên mạng.

Ứng dụng thứ nhất là phát hiện các đối tượng có khả năng là nguồn phát hay mục tiêu trong các cuộc tấn công từ chối dịch vụ. Việc cài đặt giải pháp ở hệ thống mạng phía nhà cung cấp dịch vụ hoặc phía mạng của các tổ chức, doanh nghiệp có thể giúp hạn chế các đối tượng có khả năng tấn công và cảnh báo sớm cho các nhà quản trị hệ thống mạng khách hàng để có giải pháp ứng phó kịp thời, nhằm đảm bảo các máy chủ cung cấp dịch vụ hoạt động ổn định, thông suốt.

Ứng dụng thứ hai là phát hiện các đối tượng có khả năng là sâu Internet đang quét mạng (quét không gian địa chỉ IP) nhằm phát hiện các máy bị lỗ hổng để tiến hành lây nhiễm. Trong các bước lây nhiễm sâu đối với một số loại sâu dạng "scanning worm", ở bước *phát hiện mục tiêu* các đối tượng tiến hành quét mạng để phát hiện các lỗ hổng của các thiết bị trên mạng làm ảnh hưởng đến hoạt động của mạng. Phát hiện nhanh các Hot-IP là bước quan trọng để phát hiện sớm các đối

tượng có khả năng là sâu đang quét mạng nhằm ngăn chặn hành động lây nhiễm tiếp theo của nó, giúp việc phòng chống có hiệu quả hơn.

Ứng dụng thứ ba là phát hiện các đối tượng có khả năng là các thiết bị đang hoạt động bất thường trong hệ thống. Trong quá trình hoạt động của các thiết bị như các máy chủ, thiết bị định tuyến,…của một trung tâm dữ liệu có thể xảy ra các trạng thái bất thường như hoạt động quá tải hay rơi vào tình trạng chập chờn do hỏng hóc hay các nguyên nhân khác do hậu quả của các cuộc tấn công mạng. Việc phát hiện các đối tượng này có ý nghĩa quan trọng để tiến hành nâng cấp, sửa chữa, bảo trì kịp thời.

Ứng dụng thứ tư là theo dõi, giám sát hoạt động của các Hot-IP kết hợp với một số điều kiện khác như theo dõi tài nguyên hoạt động của hệ thống để có thể hạn chế hay điều phối hoạt động của các luồng dữ liệu chứa các Hot-IP này.

4.2. PHÁT HIỆN CÁC ĐỐI TƯỢNG CÓ KHẢ NĂNG LÀ MỤC TIÊU, NGUỒN PHÁT TRONG TẤN CÔNG TỪ CHỐI DỊCH VỤ

4.2.1. Ý nghĩa thực tiễn

Tấn công DoS/DDoS là các cuộc tấn công rất nguy hiểm trên mạng bởi tính đơn giản trong việc thực hiện cuộc tấn công và hậu quả để lại rất nghiêm trọng của nó. Đặc điểm quan trọng của các cuộc tấn công này là các máy thực hiện tấn công gửi một số lượng rất lớn các gói tin yêu cầu đến các máy chủ nạn nhân làm cho các máy chủ này tiêu tốn, cạn kiệt tài nguyên và ngăn cản sự truy cập của những người dùng hợp lệ.

Phát hiện các đối tượng có khả năng là nguồn phát hay mục tiêu trong các cuộc tấn công từ chối dịch vụ có ý nghĩa quan trọng, giúp hạn chế các nguy hại trên mạng, đảm bảo các ứng dụng và dịch vụ trên mạng hoạt động ổn định, thông suốt.

4.2.2. Vấn đề nghiên cứu đặt ra

Như đã phân tích ở chương 1, có ba nhóm giải pháp trong phòng chống tấn công từ chối dịch vụ tương ứng với các giai đoạn tấn công: giai đoạn *trước khi xảy*

ra tấn công, thực hiện các giải pháp *đề phòng*; giai đoạn *trong khi xảy ra tấn công*, thực hiện các giải pháp *phát hiện và phản ứng lại tấn công*; giai đoạn *sau khi xảy ra tấn công* (hậu tấn công), thực hiện các giải pháp *"lần vết"* hay còn gọi là *"dò ngược"* để truy tìm nguồn gốc tấn công.

Các giải pháp này gồm có các loại như phân tích thống kê, học máy, khai khoán dữ liệu, dựa vào các dấu hiệu được định nghĩa sẵn với mục tiêu chính là phát hiện trong dòng dữ liệu có khả năng bị tấn công hay không. Để phát hiện nguồn phát tấn công phải thực hiện ở bước hậu tấn công với các kỹ thuật "dò ngược".

Tìm ra giải pháp cân bằng giữa phát hiện khả năng tấn công và phát hiện các đối tượng có khả năng là nguồn phát hay mục tiêu trong các cuộc tấn công có ý nghĩa quan trọng. Dựa vào bài toán phát hiện các Hot-IP có thể giải quyết vấn đề này.

Khi tấn công từ chối dịch vụ diễn ra, các nguồn phát và mục tiêu trong các cuộc tấn công này là các Hot-IP trên mạng. Mục tiêu của giải pháp là phải phát hiện các nguy cơ tấn công cũng như các nạn nhân để tiến hành ngăn chặn kịp thời. Việc này có ý nghĩa quan trọng nhằm hạn chế các tác hại của nó. Muốn vậy, giải pháp sử dụng phải có khả năng phát hiện nhanh, đơn giản và hiệu quả. Bài toán này có thể được mô hình hóa về bài toán phát hiện các Hot-IP dựa vào thử nhóm bất ứng biến.

Các phân tích ở chương 1 về bài toán phòng chống tấn công DoS/DDoS và các nghiên cứu gần đây [37][38] cho thấy phương pháp thử nhóm bất ứng biến có nhiều lợi thế trong việc phát hiện các tấn công từ chối dịch vụ và tấn công từ chối dịch vụ phân tán. Những hiệu quả chính của phương pháp này mang lại là phát hiện nhanh, đơn giản và độ chính xác cao, không cần lưu trữ các dấu hiệu tấn công, không cần thiết lập các hành vi bình thường hay bất thường hoặc kiểm tra từng yêu cầu một như trong các hệ thống IDS/IPS hay các giải pháp "dò ngược".

Khattab đề xuất giải pháp "live baiting" phát hiện tấn công từ chối dịch vụ bằng phương pháp thử nhóm bất ứng biến [37]. Nhóm tác giả này cũng đã chứng minh tính hiệu quả của giải pháp so với các giải pháp đề xuất khác. Tuy nhiên,

Khattab sử dụng ma trận d-phân-cách bằng phương pháp xác suất, do đó việc lưu trữ ma trận rất lớn sẽ dẫn đến không hiệu quả và khó triển khai áp dụng vì tài nguyên trên các thiết bị mạng vốn rất hạn chế.

Hai mô hình có thể ứng dụng việc phát hiện các đối tượng phát động tấn công và nạn nhận trong các cuộc tấn công DoS/DDoS dựa vào việc phát hiện các Hot-IP trực truyến như sau: Mô hình thứ nhất ứng dụng ở các mạng trung gian như mạng của các nhà cung cấp dịch vụ (ISP). Trong mô hình này, giải pháp có thể tích hợp vào các router trung gian để phát hiện, cảnh báo, điều phối hoạt động của các nguy cơ của các tấn công từ chối dịch vụ. Các địa chỉ IP được xem xét xử lý như nhau trong luồng lưu lượng. Mô hình thứ hai ứng dụng trong mạng của các nhà cung cấp ứng dụng hay dịch vụ trên Internet (gọi là các IsSP – Internet special Service Provider). Trong mô hình ứng dụng này, giải pháp được tích hợp vào các bộ gateway của hệ thống cung cấp dịch vụ. Các địa chỉ IP trong trường hợp này có sự phân biệt giữa các địa chỉ của những người đăng ký sử dụng dịch vụ và những địa chỉ của người dùng không đăng ký. Những địa chỉ IP không đăng ký sử dụng dịch vụ có thể dùng một số ít địa chỉ để đại diện. Điều này có ý nghĩa rất lớn trong việc lựa chọn kích thước ma trận trong giải pháp, ưu tiên sử dụng dịch vụ đối với những địa chỉ đăng ký là mục tiêu của các nhà cung cấp dịch vụ. Đối với những địa chỉ không đăng ký thì khả năng xuất hiện Hot-IP cao hơn, hệ thống có thể ngắt kết nối với các Hot-IP trong một hoặc một vài chu kỳ thuật toán để hạn chế truy cập các ứng dụng, dịch vụ cung cấp.

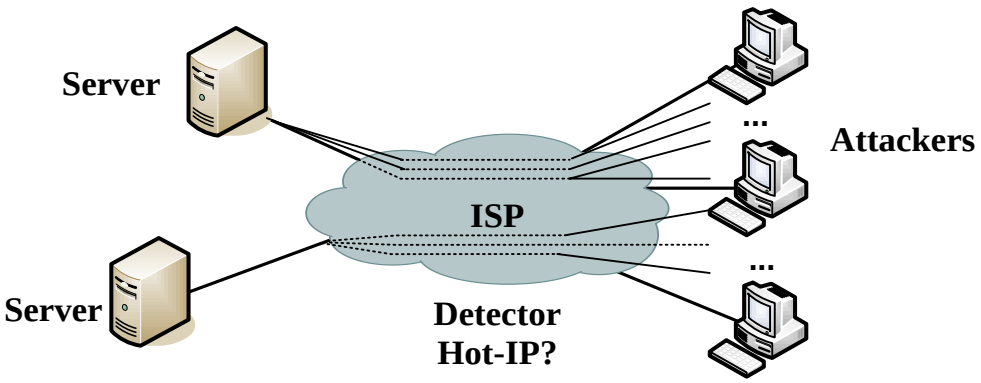
Hai bài toán đặt ra trong hai mô hình trên là: bài toán thứ nhất là phát hiện trực tuyến các IP có khả năng là nguồn phát tấn công từ chối dịch vụ (ứng dụng trong gatewate ở IsSP) hay nạn nhân trong các cuộc tấn công này (ứng dụng trong các router trung gian của ISP). Bài toán thứ hai là đảm bảo hệ thống mạng hoạt động ổn định, thông suốt, tránh quá tải của các thiết bị định tuyến trung gian hay đảm bảo hệ thống server cung cấp dịch vụ hoạt động ổn định bằng cách ngăn chặn các Hot-IP trong một hoặc một vài chu kỳ thuật toán.

Luận án sử dụng phương pháp nối mã để xây dựng ma trận này một cách tường minh. Trong đó, các cột của ma trận được phát sinh, tính toán, giảm không gian lưu trữ, tăng khả năng áp dụng vào hệ thống mạng thực tế. Giải pháp có thể triển khai được trong trường hợp các thiết bị cài đặt có tài nguyên hạn chế và đảm bảo ma trận phát sinh chính xác là ma trận phân cách.

Các thuật toán cải tiến mà luận án đã đề xuất ở chương 2 được ứng dụng vào trường hợp phát hiện nguy cơ tấn công DoS/DDoS, đồng thời phát hiện các địa chỉ có khả năng là nguồn phát hay các nạn nhân trong các cuộc tấn công này.

4.2.3. Mô hình hóa về bài toán phát hiện Hot-IP

Dựa vào đặc trưng cơ bản của một cuộc tấn công từ chối dịch vụ hay tấn công từ chối dịch vụ phân tán là các đối tượng tấn công gửi với số lượng rất lớn gói tin yêu cầu trong khoảng thời gian rất ngắn đến mục tiêu tấn công. Như vậy, xét dòng gói IP lưu thông qua hệ thống các router trung gian ở phía ISP, các gói tin được trích ra địa chỉ IP đích trong IP-header để phân tích. Nếu quan sát các gói dữ liệu đi qua router mà trong đó có rất nhiều gói có cùng đích đến thì có khả năng địa chỉ IP đó đang bị tấn công từ chối dịch vụ. Hình 4.1 mô tả các máy chủ bị các kẻ tấn công từ chối dịch vụ.



Hình 4.1. *Mô hình tấn công từ chối dịch vụ*

Gọi Δ là thời gian một chu kỳ thuật toán, $N_\Delta (N_\Delta \le N)$ là số lượng địa chỉ IP phân biệt trong khoảng thời gian Δ, m_Δ là số lượng gói tin IP hệ thống có thể nhận được trong khoảng thời gian Δ. Gọi Hot-List là danh sách lưu các địa chỉ IP nghi

ngờ trong quá trình thực thi thuật toán, kích thước của Hot-List được lựa chọn là số lượng mục tiêu trong tấn công DoS/DDoS hoặc số lượng nguồn phát tấn công DoS/DDoS giải pháp có thể phát hiện được. Ngưỡng tần suất cao $\delta = \frac{m_{\Delta}}{|\text{Hot-List}|}$.

Xây dựng ma trận d-phân-cách kích thước $t \times N$, trong đó các cột của ma trận tương ứng với các địa chỉ IP phân biệt, các dòng của ma trận tương ứng là các nhóm thử. Các gói tin IP trong dòng IP được phân tích để trích địa chỉ IP đích và ánh xạ thành các chỉ số trong [N]. Áp dụng thuật toán cải tiến 1 "Online Hot-IP Detecting" để phát hiện các đối tượng có khả năng là các mục tiêu hoặc các nguồn phát tấn công trong tấn công DoS/DDoS, giải pháp này chỉ mang tính chất phát cảnh báo cho người quản trị khi phát hiện được các Hot-IP. Thuật toán cải tiến 2 "Online Hot-IP Preventing" được ứng dụng để giữ ổn định cho hoạt động của hệ thống máy chủ bằng cách ngắt kết nối đối với các Hot-IP phát hiện được trong một hoặc một vài chu kỳ thực hiện thuật toán với mô hình giải pháp được cài đặt Inline, nghĩa là cài đặt trên thiết bị nằm trên đường truyền dữ liệu.

Các tham số của thuật toán như sau:

- Tham số đầu vào: các IP được trích ra trong IP-header của các gói tin IP trong một chu kỳ thuật toán. Chu kỳ thuật toán do người quản trị cấu hình.
- Tham số đầu ra: các IP là Hot-IP

4.2.4. Kịch bản thực nghiệm và kết quả

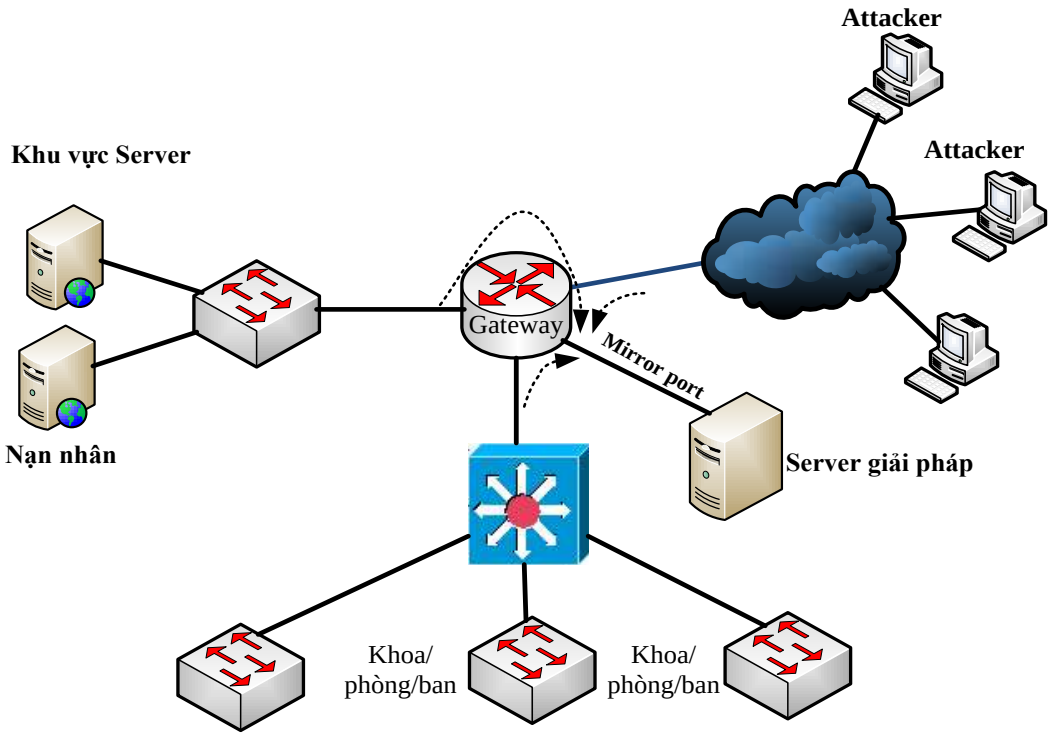
Trong phần thực nghiệm phát hiện các Hot-IP có khả năng là nguy cơ gây ra tấn công từ chối dịch vụ phân tán, chúng tôi sử dụng hệ thống mạng tại trường Đại học Sư phạm Kỹ thuật Tp.Hồ Chí Minh.

Mô hình thực nghiệm 1: Phát hiện các Hot-IP có khả năng là nguồn phát tấn công DoS/DDoS

Server cài đặt giải pháp được đặt theo dạng *"promiscuous"*, tức là ở dạng thu thập dữ liệu từ các dòng dữ liệu qua router gateway. Router gateway được cấu

hình để nhân bản các dữ liệu qua nó và gửi cho Server nhằm tránh ảnh hưởng đến dòng dữ liệu đang hoạt động. Mô hình thực nghiệm được thể hiện trong hình 4.2.

Luận án sử dụng công cụ tấn công DoSHTTP ở các máy tấn công để gửi số lượng gói tin lớn đến các server nạn nhân. Giải pháp được viết bằng C/C++ cài đặt trên HĐH CentOS (IBM X3650, Processors Intel Xeon ® CPU 2,5 GHz, RAM 4GB, Hệ điều hành CentOS 64-bit). Ma trận *d*-phân-cách được sử dụng có khả năng hỗ trợ đến 262.144 địa chỉ IP phân biệt được xây dựng từ phép nối mã.



Hình 4.2. *Mô hình mạng thực nghiệm phát hiện tấn công DDoS*

❖ *Các tham số sử dụng:*

Ma trận d-phân-cách được sinh ra từ phép nối mã với mã ngoài $[63,3]_{64}$-RS và mã trong I_{64}, có kích thước 4.032 dòng và 262.144 cột. Như vậy, lượng IP phân biệt có thể hỗ trợ tối đa là 262.144 trong dòng gói tin IP cần xử lý. Giá trị *d* được chọn là 20, nghĩa là có thể phát hiện tối đa 20 nguồn phát tấn công DoS. Thử nghiệm trong chu kỳ Δ (từ 5 giây đến 30 giây), với 10 nguồn phát tấn công DoS vào máy chủ mục tiêu.

Về chu kỳ thuật toán được chọn trong thực nghiệm dựa trên cơ sở như sau: Theo nghiên cứu của nhóm Visoottiviseth [76] về số lượng gói tin có thể xử lý trên các hệ điều hành thì mỗi nhân (core) của hệ điều hành Linux có thể xử lý tối đa khoảng 50.000 gói tin/giây. Như vậy, với server Linux thử nghiệm có cấu hình IBM Quad Core Xeon E5420 2.5 Ghz có khả năng xử lý khoảng 4x50.000 =200.000 gói tin/giây. Do đó, trong thời gian 30 giây chu kỳ thuật toán, server có thể xử lý 30 x 200.000 = 6.000.000 gói tin. Do các điều kiện về tắt nghẽn, giả sử có thể lấy giá trị bằng 2/3 so với khả năng xử lý, tức là khoảng 4.000.000 gói tin trong 30 giây.

Giả sử thời gian xử lý một gói tin "ping" bình thường giữa 2 máy tính là 10ms. Trong thời gian 30 giây, mỗi máy tính có thể gửi khoảng 3000 gói ping. Với khoảng 25 máy tính thử nghiệm đại diện cho các IP bình thường thì trong 30 giây có thể gửi khoảng 25x3.000 = 75.000 gói tin. Giả sử số lượng gói tin tấn công xuất phát từ một máy tấn công trung bình khoảng 10.000 gói/giây (qua các công cụ như đã trình bày bên trên), với 10 máy phát tấn công trong 30 giây có khoảng 3.000.000 gói tin. Do đó, tổng gói có khả năng gửi vào server khoảng 3.075.000 gói tin trong 30 giây.

Như vậy, khoảng thời gian chu kỳ thuật toán trong thực nghiệm được chọn lớn nhất là 30 giây để server thử nghiệm có khả năng xử lý được (khả năng xử lý của server trong 30 giây là 4.000.000 gói tin và số lượng gói tin trong thử nghiệm của các máy gửi đến server là khoảng 3.075.000). Do đó, các khoảng chu kỳ thuật toán được chọn trong các thử nghiệm là 5 giây, 10 giây, 15 giây, 20 giây, 25 giây, 30 giây.

Giá trị ngưỡng δ được xác định theo phương pháp của Cormode [27]. Tham số đầu vào của thuật toán là các IP được trích ra trong IP-header của các gói tin trong một chu kỳ thuật toán. Thuật toán thông báo kết quả các Hot-IP tìm được sau mỗi chu kỳ thuật hiện. Kết quả thực nghiệm được thể hiện ở bảng 4.1

Bảng 4.1. *Kết quả thực nghiệm thuật toán cải tiến 1*

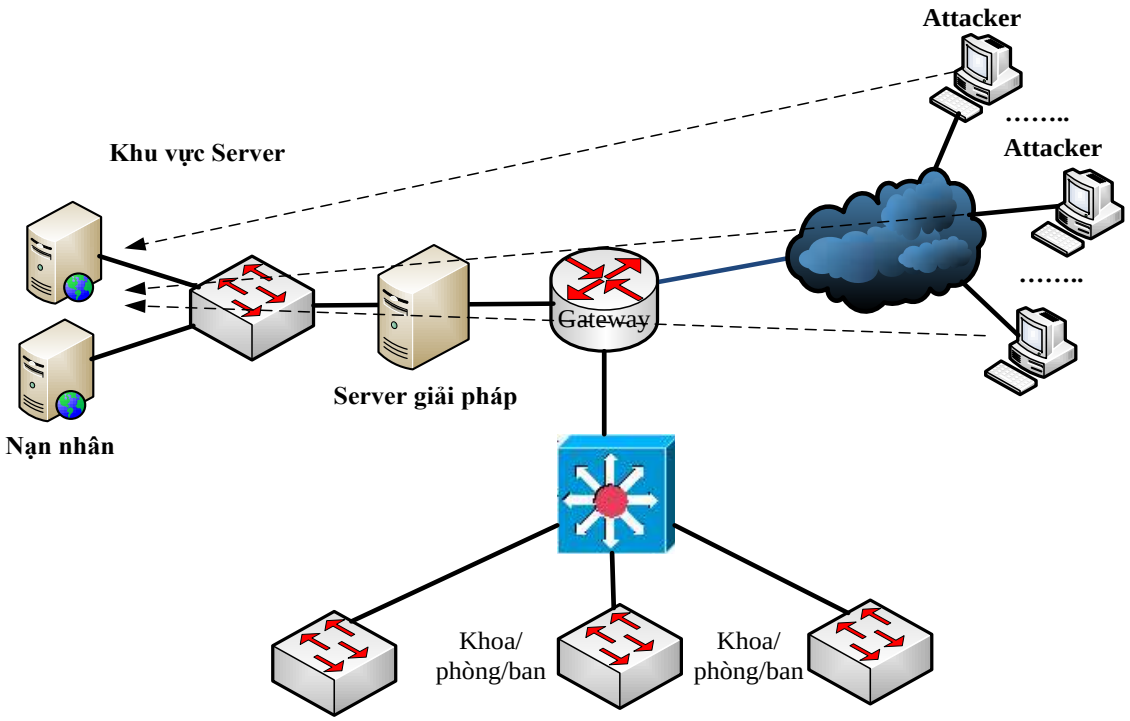
N	Δ (giây)	Tỉ lệ phát hiện Hot-IP	D	Hot-IP
100.000	$\Delta = 5$	100%	20	10
100.000	$\Delta = 10$	100%	20	10
100.000	$\Delta = 15$	100%	20	10
100.000	$\Delta = 20$	100%	20	10
100.000	$\Delta = 25$	100%	20	10
100.000	$\Delta = 30$	100%	20	10

Qua kết quả thực nghiệm cho thấy rằng giải pháp thử nhóm bất ứng biến cải tiến "Online Hot-IP Detecting" thực hiện tốt với số lượng đối tượng lớn trên mạng, thời gian giải mã để phát hiện các Hot-IP nhanh, có thể triển khai áp dụng vào thực tế.

Mô hình thực nghiệm 2: Phòng chống tấn công DDoS

Giải pháp phòng chống tấn công từ chối dịch vụ được cài đặt trên thiết bị đặt trước hệ thống máy chủ nạn nhân theo dạng Inline, sử dụng công cụ tấn công DoS đánh vào tầng ứng dụng của các máy chủ trong hệ thống làm cạn kiệt tài nguyên và mất khả năng phục vụ của máy chủ Web. Mục tiêu của thực nghiệm này là phát hiện và tạm thời ngắt kết nối đối với các Hot-IP phát hiện được (các máy phát tấn công DoS) trong một chu kỳ thực hiện thuật toán nhằm đảm bảo hệ thống hoạt động ổn định, thông suốt. Sơ đồ thực nghiệm thể hiện trên hình 4.3.

Trong phần thực nghiệm, 10 máy tính được sử dụng cài đặt phần mềm HTTP-DoS để tấn công vào Web server. Máy chủ cài đặt giải pháp thực hiện với chu kỳ thuật toán từ 5 – 30 giây để xử lý dòng dữ liệu qua nó, thuật toán được sử dụng trong trường hợp này là thuật toán cải tiến 2 "Online Hot-IP Preventing". Kết quả thực nghiệm được thống kê trong bảng 4.2.



Hình 4.3. *Sơ đồ thực nghiệm phòng chống tấn công DdoS*

Bảng 4.2. *Kết quả thực nghiệm thuật toán cải tiến 2*

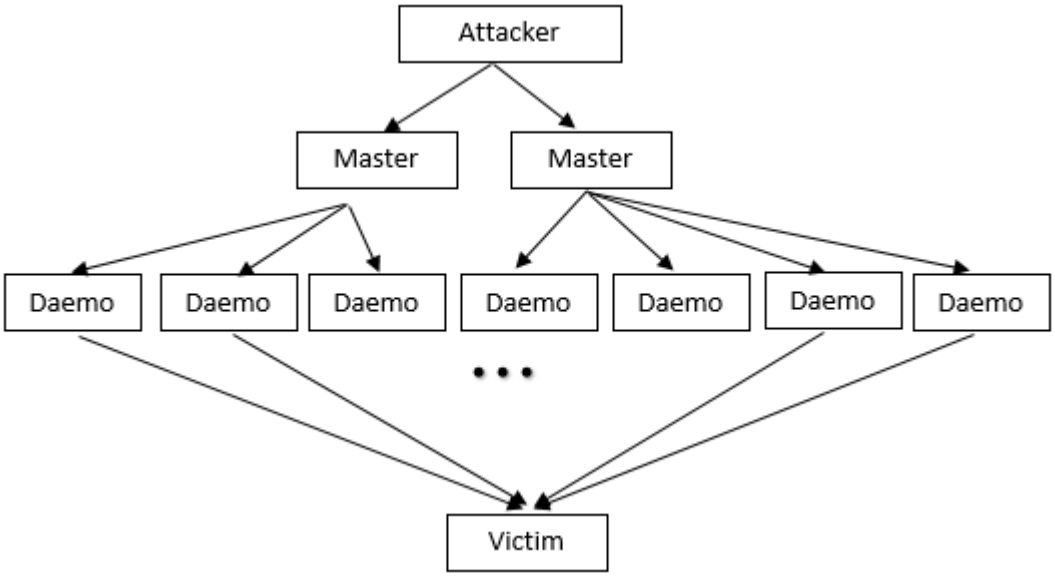
N	Δ	CPU	D	Hot-IP
100.000	$\Delta = 5$	25%	20	10
100.000	$\Delta = 10$	33%	20	10
100.000	$\Delta = 15$	45%	20	10
100.000	$\Delta = 20$	77%	20	10
100.000	$\Delta = 25$	91%	20	10
100.000	$\Delta = 30$	100%	20	10

Qua kết quả thực nghiệm cho thấy việc thiết lập thời gian cho một chu kỳ thuật toán có ý nghĩa quan trọng, giá trị này kết hợp với khả năng của vị trí triển khai, kích thước ma trận như đã đề cập ở chương trước để lựa chọn sao cho giải pháp có thể loại bỏ các nguy cơ tấn công và đảm bảo hệ thống mạng hoạt động ổn định, thông suốt.

❖ **Thực nghiệm tấn công với Trinoo:**

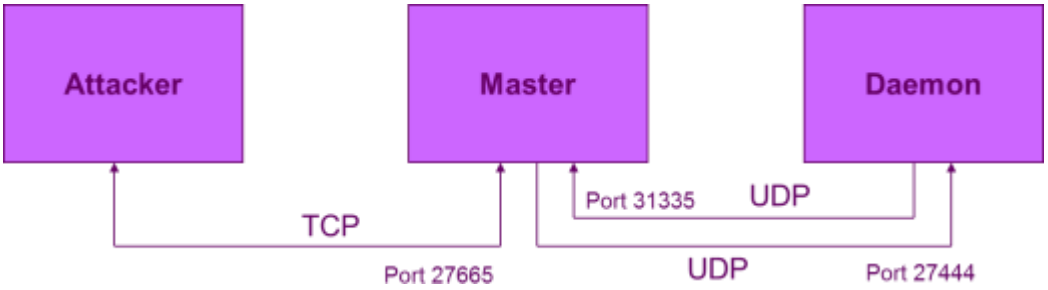
Nguyên tắc chung đối với các phương pháp phòng thủ là máy chủ ứng dụng không bị tấn công trực tiếp bằng cách sử dụng mạng các bộ phát hiện Hot-IP làm trung gian giữa người dùng và ứng dụng với số lượng lớn. Các bộ phát hiện Hot-IP trung gian có cơ chế tương tác, phối hợp linh động để phát hiện sớm các nguy cơ có thể tấn công và ngăn chặn kịp thời.

Luận án sử dụng công cụ **Trinoo** để xây dựng một hệ thống tấn công DDoS phục vụ cho việc kiểm thử chương trình. Trinoo bao gồm một chương trình phá hoại và một chương trình chủ. Chương trình chủ được sử dụng để điều khiển mạng Trinoo tạo ra các cuộc tấn công. Với mạng Trinoo này, các kịch bản tấn công DDoS được tạo ra với cường độ thay đổi bằng cách thay đổi tổng tốc độ lưu lượng tấn công. Mô hình tấn công của Trinoo để thể hiện trên hình 4.4.



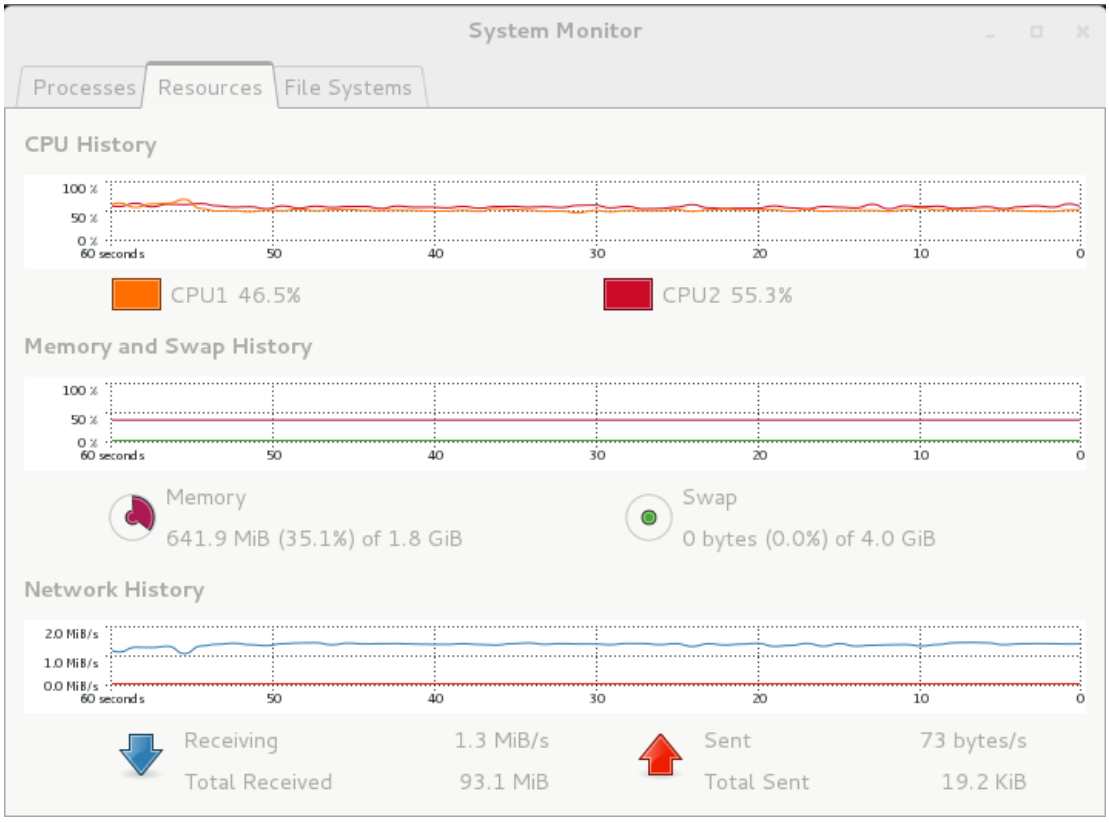
Hình 4.4. *Mô hình tấn công của Trinoo*

Mô hình triển khai thực nghiệm gồm 2 máy chủ "Master" được điều khiển bởi kẻ tấn công, có chức năng gửi các lệnh yêu cầu tấn công từ kẻ tấn công đến các "daemon" thuộc phạm vi quản lý. Sử dụng 20 máy tính client đã cài chương trình backdoor từ kẻ tấn công được gọi là các *daemon*. Các *daemon* này sẽ đồng loạt tấn công vào máy chủ nạn nhân khi nhận được một lệnh từ "Master" chuyển đến. Mô hình giao tiếp giữa các thành phần của Trinoo được thể hiện trên hình 4.5.

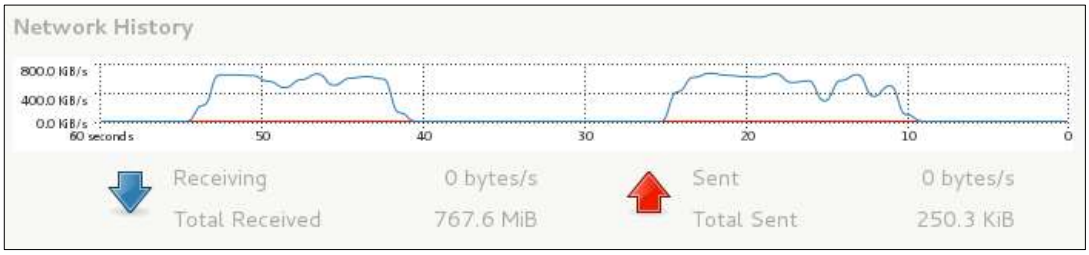


Hình 4.5. *Các cổng giao tiếp giữa các thành phần của Trinoo.*

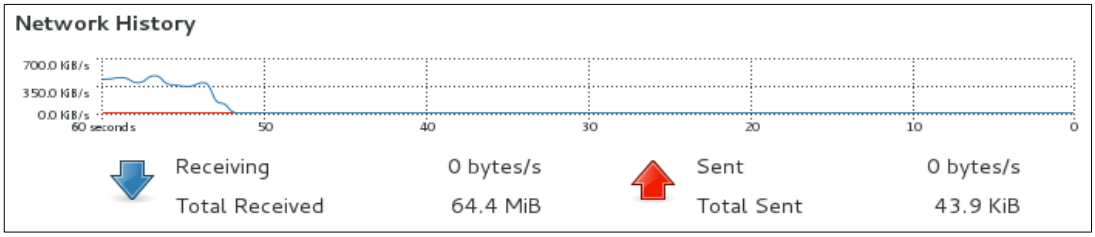
Kết quả thực nghiệm với Trinoo:



Hình 4.6. *Máy chủ nạn nhân bị tấn công*



Hình 4.7. *Các Hot-IP bị chặn trong một chu kỳ thuật toán*



Hình 4.8. *Các Hot-IP bị chặn*

Qua các thực nghiệm về ứng dụng giải pháp phát hiện các Hot-IP trực tuyến trong bài toán phát hiện các đối tượng có khả năng là mục tiêu hay nguồn phát tấn công DoS/DDoS cho thấy rằng các bộ phát hiện Hot-IP có thể cung cấp hiệu quả khả năng phục hồi trước các tấn công DoS/DDoS và bảo vệ máy chủ ứng dụng. Cụ thể, các bộ phát hiện Hot-IP có thể phát hiện các cuộc tấn công DoS/DDoS một cách nhanh chóng trong môi trường mạng lớn và có khả năng ngăn chặn các đối tượng nguy cơ (Hot-IP) trong các cuộc tấn công từ chối dịch vụ bằng cách kích hoạt tường lửa ngắt các Hot-IP này trong một chu kỳ thuật toán, đảm bảo hệ thống hoạt động ổn định.

4.3. PHÁT HIỆN CÁC ĐỐI TƯỢNG CÓ KHẢ NĂNG LÀ NGUỒN PHÁT TÁN SÂU INTERNET

4.3.1. Ý nghĩa thực tiễn

Sâu mạng hay sâu Internet là những chương trình máy tính độc hại tự phân tán bằng cách khai thác các lỗ hổng của các máy tính trên mạng. Với khả năng tự sao chép và lan truyền, sâu Internet là kiểu tấn công Internet quy mô lớn.

Một trong những bước đầu tiên của việc phát tán sâu là quét mạng với tốc độ cao. Chúng tập hợp những danh sách với hàng ngàn địa chỉ IP và quét mạng với tốc độ cao để tìm kiếm các máy bị lỗ hổng để khai thác. Code Red v2 khai khác lỗ hổng mạng qua IIS lây nhiễm khoảng 360.000 máy [56], Sapphire khai thác lỗ hổng SQL server và lây nhiễm 75.000 máy [57]. Vấn đề được đặt ra là làm thế nào để phát hiện ra chúng càng nhanh càng tốt.

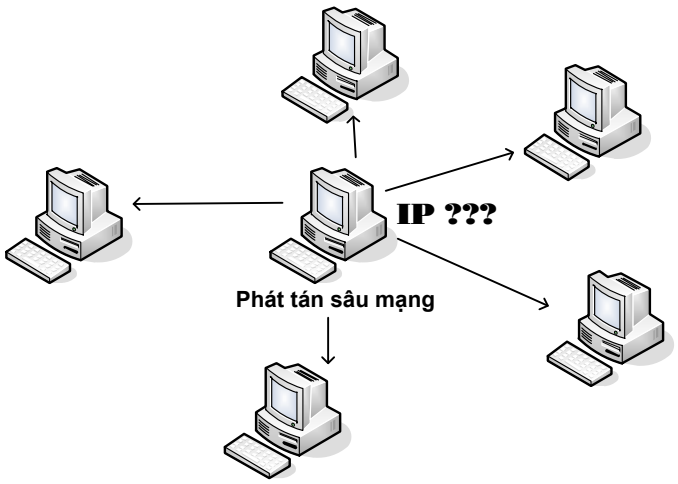
Đối với các cuộc tấn công sâu Internet, nhiều kỹ thuật bảo vệ khác nhau đã được đề xuất để ngăn chặn sự lan truyền sâu. Các kỹ thuật phát hiện sớm các sâu mạng phổ biến như: nhóm nghiên cứu của Zou [20] đề xuất giải pháp phát hiện dựa vào bộ lọc Kalman. Phương pháp này dò tìm các xu hướng quét mạng bất hợp pháp qua một số lượng lớn các IP không sử dụng. Nhóm nghiên cứu của Wu [58] dựa vào bộ đếm theo dõi tỷ lệ tăng của các máy bị nhiễm mới. Các sâu mạng được cảnh báo khi các sự kiện bất thường xảy ra vượt qua một ngưỡng xác định nào đó. Nhóm nghiên cứu của Berk [59] sử dụng phương pháp giám sát hệ thống bằng cách thống kê các gói tin ICMP *"Destination Unreadable"* trên router đối với các gói tin đến các IP không sử dụng.

Đặc điểm quan trọng của sâu Internet là việc phát tán cực nhanh trên mạng ở bước quét mạng để tìm kiếm mục tiêu. Nghĩa là nó gửi một số lượng rất lớn gói tin trong một khoảng thời gian rất ngắn đến các địa chỉ đích khác nhau. Mục đích của việc này là dò tìm lỗ hổng của các máy trên mạng để lây nhiễm. Chúng ta có thể bắt gói thông qua router và phân tích để xác định các địa chỉ IP gửi với tần suất cao này (các Hot-IP) trên mạng. Phương pháp này cũng có thể phát hiện các sâu chưa được biết đến.

4.3.2. Vấn đề nghiên cứu đặt ra

Trong trường hợp phát tán sâu Internet, máy phát tán sâu phát đi đến các địa chỉ khác trong mạng. Nếu quan sát trên luồng dữ liệu trên mạng có quá nhiều gói có cùng địa chỉ nguồn thì máy nguồn này có thể đang bị nhiễm sâu và nó đang quét mạng. Hình 4.9 mô tả máy nhiễm sâu đang quét không gian địa chỉ IP để tìm kiếm các máy có lỗ hổng để phát tán sâu.

Có thể mô hình hóa bài toán phát hiện các đối tượng có khả năng là sâu Internet đang quét mạng về bài toán phát hiện các Hot-IP để phát hiện, ngăn chặn và cảnh báo sớm nhằm giúp hạn chế quá trình lây lan của sâu. Dựa vào phân tích các gói tin IP trực tuyến và phát hiện các Hot-IP chính là phát hiện các máy có khả năng nhiễm sâu đang tiến hành quét mạng.



Hình 4.9. *Máy nhiễm sâu đang phát tán trên mạng*

4.3.3. Mô hình hóa về bài toán phát hiện Hot-IP

Gọi Δ là chu kỳ thực hiện thuật toán, m_Δ là số lượng gói tin tối đa mà hệ thống có thể nhận được trong khoảng thời gian Δ, Hot-List là danh sách chứa các địa chỉ IP nghi ngờ là nguồn phát tán sâu Internet (Hot-IP) trong quá trình thực hiện thuật toán, kích thước của Hot-List lớn hơn *d* và được lựa chọn tùy thuộc vào thực tế và kinh nghiệm của người quản trị.

Ngưỡng tần suất cao $\delta = \frac{m_\Delta}{|\text{Hot-List}|}$, ma trận nhị phân được sinh ra dựa vào giá trị N trong khoảng thời gian Δ của vị trí triển khai giải pháp và phép nối mã C_{in} và C_{out}.

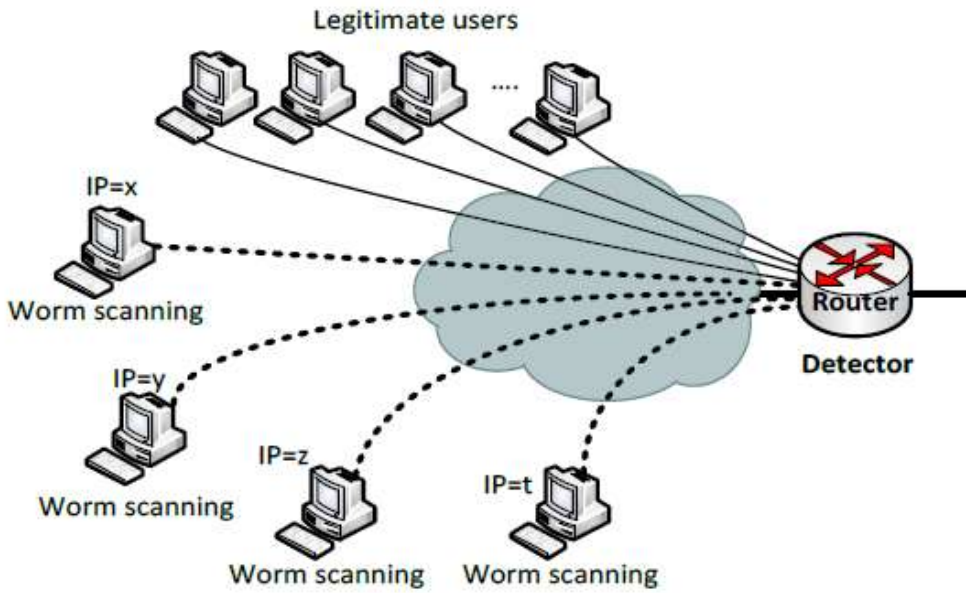
Các tham số sử dụng trong thuật toán:

- Tham số đầu vào: các IP nguồn được trích ra từ IP-header trong các gói tin IP.
- Tham số đầu ra: các IP là Hot-IP

  Các Hot-IP này có khả năng là nguồn phát tán sâu mạng.

4.3.4. Kịch bản thực nghiệm và kết quả

Gọi các máy tính nhiễm sâu đang tiến hành quét không gian địa chỉ IP để phát hiện lỗ hổng của các thiết bị trên mạng nhằm tiến hành phát tán và lây nhiễm là các Hot-IP. Luận án sử dụng phương pháp thử nhóm bất ứng biến để phát hiện các Hot-IP này bằng cách cài đặt thuật toán vào router, bắt gói và phân tích dựa vào địa chỉ IP nguồn trong các gói tin gửi đến router. Mô hình thực nghiệm phát hiện các máy nhiễm sâu đang tiến hành quét mạng được mô tả trên hình 4.10.



Hình 4.10. *Mô hình thực nghiệm phát hiện các máy nhiễm sâu trên mạng*

Giả sử rằng có *N* máy tính gửi gói tin đến router, các máy tính nhiễm sâu đang tiến hành quét mạng chính là các Hot-IP cần tìm. Giả sử trong một chu kỳ thuật toán, dòng dữ liệu được giám sát, router nhận tổng cộng *m* gói tin từ *N* máy tính trên mạng, trong đó giả sử có nhiều nhất *d* máy tính là nhiễm sâu (Hot-IP) đang tiến hành quét không gian địa chỉ IP để tìm kiếm lỗ hổng của các máy tính trên mạng để phát tán.

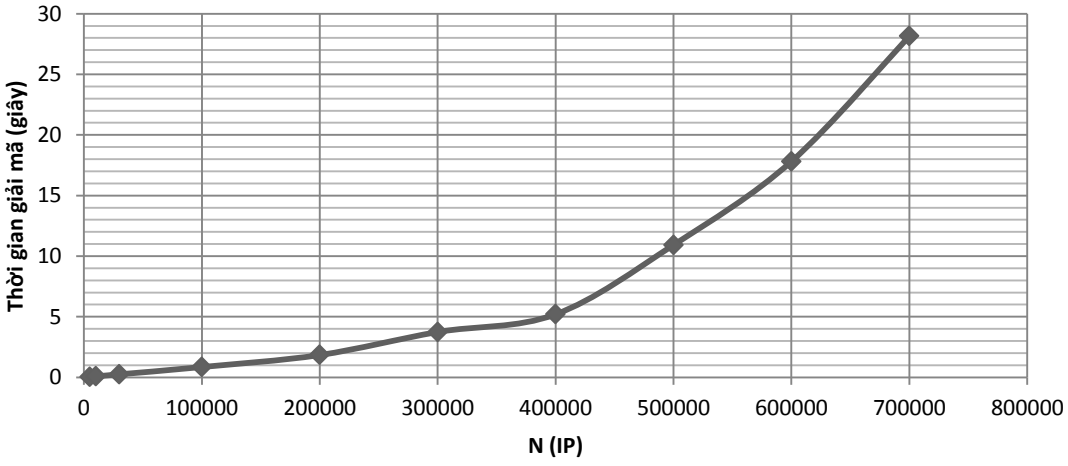
Giả sử rằng có *t* phép thử, xây dựng ma trận d-phân-cách bằng phương pháp nối mã như đã trình bày trong phần trước. Áp dụng thuật toán phát hiện các Hot-IP để tìm ra các Hot-IP chính là các máy tính đang phát tán sâu trên mạng.

❖ **<u>Kịch bản 1:</u>** Thực nghiệm này để đo thời gian giải mã phát hiện các đối tượng có khả năng là nguồn quét mạng tìm kiếm lỗ hổng để truyền sâu với khả năng lên đến 700.000 đối tượng (IP) phân biệt.

Ma trận nhị phân có kích thước 992x1000000 được phát sinh từ mã Reed Solomon C_{out}:[31, 5] và C_{in}: I_{32} (t = 31 x 32 = 992, $d = \left\lfloor \frac{31-1}{5-1} \right\rfloor = 7$). Kết quả giải mã được trình bày trong bảng 4.3.

Bảng 4.3. *Kết quả dò tìm các Hot-IP trên mạng*

N (IP)	Thời gian giải mã (giây)	N (IP)	Thời gian giải mã (giây)
5.000	0,04	300.000	3,74
10.000	0,08	400.000	5,20
30.000	0,25	500.000	10,95
100.000	0,85	600.000	17,81
200.000	1,84	700.000	28,15



Hình 4.11. *Biểu đồ mô tả thời gian giải mã phát hiện các sâu mạng*

❖ **Kịch bản 2:** Xử lý dòng dữ liệu thời gian thực

Thực nghiệm này được cài đặt với chu kỳ thuật toán $\Delta = 15$ giây, $\Delta = 20$ giây và $\Delta = 30$ giây. Thời gian thực thi thuật toán được tính toán gồm quá trình tiền xử lý (bắt gói, trích địa chỉ IP nguồn trong gói tin IP) và thời gian giải mã để phát hiện các Hot-IP.

Số lượng IP phân biệt (N) sử dụng trong thực nghiệm là 4096, ma trận nhị phân d-phân-cách được sinh ra từ phép nối mã RS $[15, 3]_{16}$ và I_{16}. Kích thước Hot-List sử dụng là 1000, nghĩa là giải pháp được cài đặt trong trường hợp này có thể phát hiện tối đa 1000 đối tượng có khả năng là sâu Internet đang tiến hành quét không gian địa chỉ nhằm tìm kiếm các mục tiêu bị lỗ hổng để truyền sâu.

Trong thực nghiệm này, sâu Internet xuất phát từ một nguồn và quét không gian địa chỉ của mạng 172.16.0.0/16 để tìm kiếm lỗ hổng của các máy trên mạng để lây nhiễm. Khi các máy bị lây nhiễm, các máy này tiếp tục quét và lây lan cho các máy khác.

Giải pháp phát hiện, hạn chế các Hot-IP có khả năng là sâu mạng đang quét không gian địa chỉ IP để phát hiện lỗ hổng và phát tán trong một chu kỳ thuật toán nhằm mục đích: thứ nhất là phát hiện và cảnh báo các máy tính có khả năng là nguồn phát tán sâu ở bước quét không gian địa chỉ để tìm kiếm mục tiêu của nó (sử dụng thuật toán cải tiến 1 – "Online Hot-IP Detecting"); thứ hai là hạn chế tốc độ lây lan bằng cách ngắt kết nối đối với các Hot-IP này trong một chu kỳ thuật toán (sử dụng thuật toán cải tiến 2 - "Online Hot-IP Preventing").

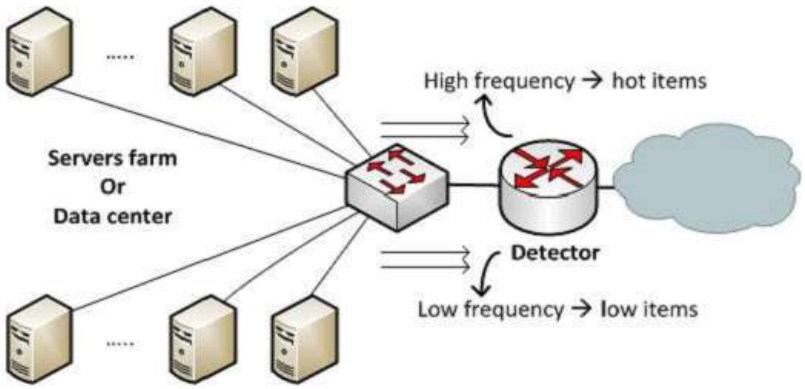
Qua kết quả thực nghiệm cho thấy rằng thời gian phát hiện các nguồn phát tán sâu mạng nhanh, hệ thống hoạt động ổn định do các nguồn phát tán bị ngăn chặn trong một chu kỳ thuật toán, có thể áp dụng triển khai vào hệ thống mạng thực tế trên cơ sở phân tích đặc điểm hệ thống mạng tại vị trí triển khai.

4.4. PHÁT HIỆN CÁC THIẾT BỊ CÓ KHẢ NĂNG HOẠT ĐỘNG BẤT THƯỜNG

4.4.1. Ý nghĩa thực tiễn

Các bất thường trên mạng là một chủ đề được nghiên cứu trong thời gian dài cho đến nay trong lĩnh vực mạng máy tính. Các bất thường có thể là tình trạng hoạt động của các máy chủ trong hệ thống hoạt động quá mức (có thể đang bị tấn công) hoặc mức độ phục vụ chập chờn dưới mức bình thường. Để theo dõi và phát hiện hiện tượng bất thường của các thiết bị trên mạng như các máy chủ, các thiết bị định tuyến, tường lửa trên mạng… thông thường người quản trị sử dụng giải pháp phổ biến là hệ thống giám sát mạng. Các hệ thống giám sát này thông thường được thiết lập trong mạng nội bộ. Các thông tin giám sát được gửi từ các thiết bị được giám sát về thiết bị giám sát, từ đó phân tích và báo cáo kết quả hoặc phát cảnh báo nếu vượt qua ngưỡng được người quản trị thiết lập trước.

Trường hợp các router hay các server phải tiếp nhận và xử lý các gói tin gửi đến quá lớn trong một khoảng thời gian ngắn như vậy có thể gọi là các "*Hot-item"* hay *"Hot-IP"*. Ngược lại, trong trường hợp các thiết bị hoạt động chập chờn có thể do hậu quả của các cuộc tấn công mạng hay có thể bị hư hỏng nào đó làm quá tải, tắc nghẽn hay đụng độ gây nên. Trường hợp này, có thể gọi đó là các *"Low item"* hay *"Low-IP"*.



Hình 4.12. *Phát hiện các thiết bị hoạt động bất thường trên mạng*

4.4.2. Vấn đề nghiên cứu đặt ra

Có nhiều nghiên cứu trong việc phát hiện các bất thường trên mạng như phương pháp thống kê [60], học máy [61], khai phá dữ liệu [62], lý thuyết thông tin [63], các nghiên cứu về hệ thống giám sát mạng. Hệ thống giám sát này kiểm soát các bất thường trên mạng như trạng thái hoạt động của các thiết bị, các dòng lưu lượng trên các cổng của thiết bị. Hệ thống sẽ phát cảnh báo khi có sự thay đổi trạng thái hoạt động hoặc khi các hoạt động vượt qua một ngưỡng đặt trước nào đó. Các phương pháp này thông thường trải qua ba bước như sau.

Bước 1. Tiền xử lý để lọc các dữ liệu đầu vào

Bước 2. Phân tích thống kê. Dữ liệu được phân loại thành các loại dữ liệu bình thường, hành vi bất thường và nhiễu.

Bước 3. Quyết định. Xác định xem có bất thường xảy ra hay không dựa trên các thông số độ lệch.

Hệ thống giám sát được sử dụng phổ biến để hỗ trợ việc giám sát tình trạng hoạt động của mạng, các thiết bị trong hệ thống nội bộ. Giao thức được dùng phổ biến trong cơ chế giám sát này là SNMP để giao tiếp giữa máy server giám sát và các thiết bị cần giám sát.

Bài toán đặt ra là trong môi trường mạng ở phía các nhà cung cấp dịch vụ làm sao có thể giám sát để cảnh báo sớm cho khách hàng rằng các server của họ có những bất thường về mặt truy cập đang diễn ra hay không trong khi ở phía nhà cung cấp dịch vụ không được phép thiết lập các cơ chế giám sát như vừa trình bày ở phần trên.

Bài toán phát hiện các thiết bị có khả năng đang hoạt động bất thường trên mạng có thể được giải quyết bằng giải pháp phát hiện các Hot-IP trực tuyến dựa trên phương pháp thử nhóm bởi những lý do như sau:

- Việc phân tích các gói tin ở tầng mạng, dựa vào địa chỉ IP thực hiện đơn giản hơn các phương pháp khác.

- Phương pháp thử nhóm tối ưu hơn về không gian lưu trữ dữ liệu xử lý do không phải tốn không gian lưu toàn bộ ma trận các phần tử cần xử lý.
- Phương pháp thử nhóm không thực hiện việc kiểm tra từng đối tượng mà gom thành từng nhóm thử nên việc xử lý sẽ nhanh hơn rất nhiều so với các phương pháp khác

4.4.3. Mô hình hóa về bài toán phát hiện Hot-IP

❖ ***Phát biểu bài toán:***

Giả sử tổng tần suất xuất hiện của *N* IP phân biệt trên dòng dữ liệu là *S* và có nhiều nhất là *d* IP có tần suất hoạt động bất thường (*Hot-IP* hoặc *Low-IP*). Một IP được coi là bình thường nếu tần suất xuất hiện của nó nhỏ hơn $\frac{S}{d+1}$ và lớn hơn $\frac{S}{(N-2d+1)(d+1)}$. Giả sử chúng ta có nhiều nhất là *d Hot-IP* và *d Low-IP* nên các IP bình thường trong dòng dữ liệu là $N-2d$. Tổng các tần suất xuất hiện của chúng lớn hơn $(N-2d).\frac{S}{(N-2d+1)(d+1)}$. Do đó, tổng các tần suất của các IP *Low-IP* nhỏ hơn $\frac{S}{(N-2d+1)(d+1)}$.

- **Phát hiện các Hot-item**: Sử dụng phương pháp của Cormode và Muthukrishnan đề xuất năm 2005 [27]. Vector nhị phân kết quả được tính toán dựa vào bộ đếm *C* tương ứng cho từng phép thử và sử dụng luật sau đây:
  - Nếu $C(i) > \frac{S}{d+1}$ thì phép thử thứ *i* có kết quả là 1
  - Ngược lại, nếu $C(i) \le \frac{S}{d+1}$ thì kết quả phép thử thứ *i* có kết quả là 0

Sau bước tính toán vector kết quả này, áp dụng thuật toán giải mã đã trình bày trong phần trước để xác định các IP là *Hot-IP* trong dòng dữ liệu. Chúng tôi ký hiệu tập các *Hot-IP* là $H = \{j_1, ..., j_h\}, \ |H| \le d.$

- **Phát hiện các Low-item:** để xác định các *Low-IP*, trước hết phải loại bỏ các *Hot-IP*.

$$CT = C - \frac{S}{d+1} \sum_{i \in H} M_j$$

Sau đó, chuyển đổi từ *CT* ra vector kết quả cho từng phép thử như sau:

- Nếu $CT(i) > \frac{S}{(N-2d+1)(d+1)}$, thì phép thử thứ *i* là 0

- Nếu $CT(i) \le \frac{S}{(N-2d+1)(d+1)}$, thì kết quả phép thử thứ *i* là 1.

Sau đó, áp dụng phương pháp giải mã để phát hiện ra các *Low-IP*.

Các tham số sử dụng trong thuật toán:

- Tham số đầu vào: các IP nguồn được trích ra từ IP-Header của các gói tin
- Tham số đầu ra: các Hot-IP và Low-IP

4.4.4. Kịch bản thực nghiệm và kết quả

- *Mô tả thực nghiệm:*

Trong phần thực nghiệm phát hiện các thiết bị có khả năng đang hoạt động bất thường trên hệ thống mạng, chúng tôi sử dụng một server (IBM X3650, Processors Intel Xeon ® CPU 2,5 GHz, RAM 4GB, hệ điều hành CentOS 64-bit) hoạt động như một router trong mạng tiếp nhận các gói tin trong dòng gói IP gửi tới nó. Thuật toán bắt gói được cài đặt bằng ngôn ngữ C, sử dụng thư viện *pcap* để phân tích các gói tin. Khi một gói tin gửi đến, phần IP-header được phân tích và rút trích thông tin về địa chỉ IP nguồn trong đó. Các địa chỉ IP này được đánh chỉ số.

Ma trận d-phân-cách được sinh ra bằng phương pháp nối mã với kích thước được tính toán dựa vào lưu lượng mạng và khả năng của hệ thống tại vị trí triển khai. Trong phần thực nghiệm này, các ma trận được sinh ra từ các mã Reed-Solomon $[7,3]_8$-RS (d=3, N=512, t=56), $[15,3]_{16}$ – RS (d=7, N=4096, t=240), $[31,3]_{32}$ – RS(d=15, N=32.768, t=992). Chúng tôi thực nghiệm trong nhiều trường hợp với số lượng *Hot-IP* và *Low-IP* khác nhau. Dựa vào ma trận d-phân-cách và tần suất xuất hiện của các địa chỉ IP trong dòng gói tin IP, vector bộ đếm cho từng phép thử được cập nhật. Từ đó, vector kết quả được tính toán dựa vào vector bộ đếm và ngưỡng được thiết lập. Kết quả về thời gian được mô tả trong bảng 4.4 và thuật toán tìm ra chính xác các Hot-IP.

- *Kết quả thực nghiệm:*

Thời gian giải mã phát hiện các *Hot-IP* và *Low-IP* được mô tả trong bảng 4.4. Qua đây, chúng ta thấy rằng giải pháp giải pháp phát hiện các Hot-IP có thể triển khai áp dụng trong hệ thống mạng thực tế để phát hiện các thiết bị có khả năng đang hoạt động bất thường trong hệ thống.

Bảng 4.4. *Thời gian giải mã phát hiện các Hot-IP và Low-IP*

RS code	Số lượng IP	Hot (Low) IP	Thời gian giải mã tìm Hot-IP (giây)	Thời gian giải mã tìm Low-IP (giây)
$[7,3]_8$	512	3	0,00	0,00
$[15,3]_{16}$	4.096	7	0,11	0,12
$[31,3]_{32}$	32.768	15	3,65	3,87

4.5. GIÁM SÁT CÁC HOT-IP

4.5.1. Ý nghĩa thực tiễn

Giám sát hoạt động của các thiết bị quan trọng trên mạng, đặc biệt là các server, là một trong những nhiệm vụ trọng tâm đối với người quản trị mạng. Mục đích của công việc này là theo dõi tình trạng hoạt động của chúng trong hệ thống

nhằm phát hiện kịp thời các bất thường về hoạt động của các thiết bị, dịch vụ, lưu lượng trên mạng. Người quản trị có thể thiết lập các ngưỡng để phát hiện và cảnh báo các trạng thái bất thường đang xảy ra. Trong hệ thống giám sát mạng gồm hai thành phần chính: máy giám sát và các thiết bị được giám sát. Máy giám sát là nơi tiếp nhận các thông tin từ các thiết bị được giám sát, phân tích và cảnh báo cho người quản trị mạng. Các thiết bị được giám sát thông thường là các thiết bị quan trọng trong hệ thống mạng như router, switch trung tâm, các máy chủ, firewall…

Giao thức thường được sử dụng trong ứng dụng này là SNMP, các thiết bị được giám sát phải được cài đặt và cấu hình để chuyển các thông báo tình trạng hoạt động cho máy chủ giám sát theo định kỳ hoặc khi nào có hoạt động bất thường xảy ra. Thời gian định kỳ này được người quản trị thiết lập trước. Giải pháp này thường áp dụng trong việc quản trị hệ thống mạng nội bộ của các tổ chức đối với các thiết bị do tổ chức đó quản lý.

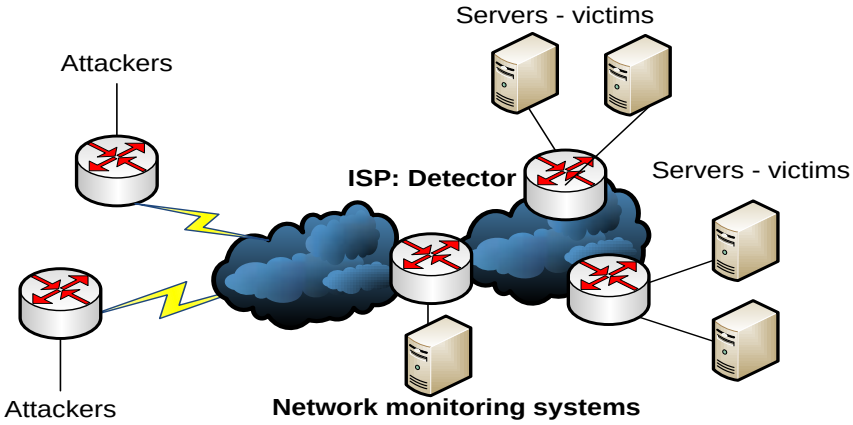
Việc theo giám sát các Hot-IP trong một số chu kỳ thuật toán là một bước quan trọng để giúp người quản trị xem xét tần suất xuất hiện của các Hot-IP này để có những nhận định, giải pháp ứng phó kịp thời trước khả năng có thể hệ thống đang bị tấn công. Dựa vào một số thông tin về tài nguyên hệ thống, người quản trị có thể thiết lập ngưỡng để hạn chế hay ngăn chặn các Hot-IP này nhằm đảm bảo hệ thống hoạt động ổn định, tránh trường hợp hệ thống bị tê liệt trước các nguy cơ bị tấn công từ chối dịch vụ.

4.5.2. Vấn đề nghiên cứu đặt ra

Vấn đề đặt ra cho việc giám sát ở các nhà cung cấp dịch vụ là theo dõi và phát cảnh báo cho khách hàng trong trường hợp các máy chủ của khách hàng có những bất thường xảy ra. Trong trường hợp này, giả sử phía nhà cung cấp dịch vụ không được phép cài đặt chế độ giám sát đối với các thiết bị của khách hàng, vì hệ thống của khách hàng mang tính chất cục bộ, thông thường sẽ không muốn rò rỉ thông tin ra bên ngoài. Việc phát hiện các Hot-IP ở phía nhà cung cấp dịch vụ có ý nghĩa quan trọng để phát hiện nhanh các khả năng nguy hại trên mạng. Bước giám

sát các Hot-IP có thể được xem là bước kế tiếp để có sự chẩn đoán chính xác hơn các nguy hại này và tiến hành cảnh báo, hạn chế tác hại hay ngăn chặn để giúp an toàn hơn cho hệ thống mạng khách hàng. Hình 4.13 mô tả mô hình giám sát các Hot-IP đặt ở phía mạng trung gian.

Trường hợp khác có thể cần phải giám sát các Hot-IP là người quản trị hệ thống muốn đảm bảo hệ thống của mình tránh các nguy cơ tấn công từ chối dịch vụ nhằm đảm bảo hệ thống server cung cấp dịch vụ ổn định. Do đó, việc theo dõi các Hot-IP kết hợp với theo dõi tài nguyên hệ thống thời gian thực có thể thiết lập các mức hạn chế hay ngăn chặn các Hot-IP là cần thiết.



***Hình 4.13**. Mô hình giám sát các Hot-IP*

4.5.3. Kịch bản thực nghiệm và kết quả

Trong phần thực nghiệm này, tác giả sử dụng 01 máy chủ Web server, 03 máy tính tấn công, 35 máy tính đóng vai trò là máy tính của người dùng bình thường, 01 máy server cài đặt giải pháp phát hiện và giám sát các Hot-IP được đặt trước máy Web server đểm kiểm soát luồng dữ liệu truy cập đến Web server.

Các máy tính tấn công sử dụng phần mềm tấn công HTTP-DoS để tấn công vào Web server, các máy tính đóng vai trò là máy tính của người dùng bình thường sử dụng công cụ trình duyệt Web để truy cập vào website trên Web server.

Các Hot-IP được phát hiện và được đưa vào trạng thái giám sát, xác định tần suất xuất hiện của chúng trong một khoảng thời gian định trước, thể hiện trên đồ thị

trực quan giúp người quản trị trong việc theo dõi hoạt động của chúng. Người quản trị có thể đặt ngưỡng để phát cảnh báo hay điều phối các luồng lưu lượng của dòng gói IP chứa các Hot-IP này.

Tần suất hoạt động quá lớn của các Hot-IP có thể làm cho cho các thiết bị định tuyến cạn kiệt tài nguyên, gây ảnh hưởng đến việc vận chuyển các luồng dữ liệu khác. Qua việc theo dõi trạng thái hoạt động của các thiết bị định tuyến, chuyển mạch trên mạng, hệ thống giám sát có thể kích hoạt tường lửa để hạn chế hay ngăn chặn các Hot-IP này. Một số tập luật có thể đưa ra như sau:

(1). ***Event****: hoạt động của CPU trong khoảng* θ_1 *và* θ_2

Condition*: true*

Action*: hạn chế tần suất Hot-IP*

(2). ***Event****: hoạt động của CPU vượt qua ngưỡng* θ_2

Condition*: true*

Action*: ngăn chặn Hot-IP*

Kết quả thực nghiệm:

Hình 4.14. *Giám sát các Hot-IP trên mạng*

Đồ thị trên cho thấy rằng tần suất hoạt động trực quan của các Hot-IP, chúng ta có thể thiết lập các chính sách nhằm kiểm soát hoạt động của chúng nhằm đảm bảo cho hệ thống làm việc tốt hơn.

Hình 4.15. *Tần suất của Hot-IP được giới hạn khi CPU trong khoảng 60%-80%*

4.6. KẾT LUẬN CHƯƠNG 4

Phát hiện và xác định các đối tượng có khả năng là các nguy cơ gây nên các cuộc tấn công từ chối dịch vụ, phát tán sâu Internet, hay các thiết bị có khả năng đang hoạt động bất thường, các nạn nhân trong các cuộc tấn công từ chối dịch vụ trên mạng giúp cảnh báo sớm cho người quản trị nhằm hạn chế các nguy hại của chúng có thể gây ra có ý nghĩa quan trọng trong lĩnh vực an ninh mạng. Trong chương này, luận án đã trình bày việc mô hình hóa các bài toán trên về dạng bài toán phát hiện các Hot-IP trên mạng.

Phương pháp phát hiện các Hot-IP dựa trên thử nhóm bất ứng biến có nhiều ưu điểm về tính đơn giản, nhanh chóng và chính xác để triển khai trong môi trường thực tế. Các mô hình xử lý song song và mô hình phân tán có thể áp dụng kết hợp để nâng cao khả năng của giải pháp và phù hợp với các mô hình mạng ngày nay, đặc biệt có ý nghĩa quan trọng trong các hệ thống mạng lớn như các ISP. Các nội dung chính cảu chương này được công bố trong các công trình [C2][C3][C5][C8].

KẾT LUẬN

Luận án trình bày giải pháp phát hiện các Hot-IP trực tuyến trên mạng dựa trên phương pháp thử nhóm bất ứng biến. Mục tiêu của chính của luận án là đề xuất giải pháp phát hiện nhanh các Hot-IP trên mạng, ứng dụng trên mạng trung gian ở các nhà cung cấp dịch vụ hoặc các mạng cung cấp dịch vụ trên Internet nhằm giúp người quản trị phát hiện nhanh, ứng phó kịp thời với các khả năng là nguy cơ ảnh hưởng xấu đến hoạt động của mạng và đảm bảo hệ thống hoạt động ổn định, thông suốt.

Bài toán phát hiện các Hot-IP có ý nghĩa quan trọng trong việc phát hiện sớm các đối tượng có khả năng gây nguy hại trên mạng. Trong các giải pháp đã khảo sát, giải pháp phát hiện các Hot-IP sử dụng thử nhóm bất ứng biến là giải pháp hữu hiệu để triển khai áp dụng nhằm phát hiện trực tuyến và hạn chế hoạt động của các đối tượng có khả năng là nguy cơ gây hại hoặc mục tiêu tấn công trên mạng. Đây là giải pháp có tính đơn giản, chính xác, tính toán nhanh, phù hợp để áp dụng trên môi trường mạng có số lượng người dùng rất lớn.

Luận án đã mô hình hóa bài toán phát hiện các Hot-IP dựa vào phương pháp thử nhóm bất ứng biến. Trong đó, luận án giải quyết vấn đề xây dựng cơ sở lý thuyết cho bài toán với các khái niệm, định nghĩa, lựa chọn các tham số đáp ứng các tiêu chí của bài toán.

Một số cải tiến giải pháp phát hiện các Hot-IP trực tuyến được luận án đề xuất để tăng tốc độ tính toán, tính chính xác và khả năng mở rộng của giải pháp. Trong đó, danh sách các địa chỉ IP nghi ngờ "Hot-List" được sử dụng với kích thước có thể mở rộng để đáp ứng cho việc phát hiện số lượng các Hot-IP lớn, giảm sự phụ thuộc và tham số d trong ma trận d-phân-cách.

Luận án cũng đã trình bày một số kỹ thuật có thể kết hợp để nâng cao khả năng của giải pháp trong việc triển khai thực tế. Trong đó, trước hết là việc lựa chọn kích thước ma trận phù hợp với khả năng của vị trí triển khai. Cụ thể đó chính là lựa

chọn các tham số dựa trên sự phân tích về chu kỳ thực hiện thuật toán, số lượng gói tin tối đa mà hệ thống xử lý được trong thời gian chu kỳ thuật toán, số lượng IP tối đa dựa vào năng lực thiết bị hay số lượng khách hàng đăng ký sử dụng dịch vụ và các địa chỉ IP đại diện.

Các cuộc tấn công mạng ngày nay có tính phối hợp cao, phân tán rộng trên Internet, để phát hiện và phòng chống hiệu quả thì chiến lược phát hiện và phòng chống cũng cần được triển khai phân tán, hợp tác giữa các thành phần. Giải pháp phát hiện các Hot-IP kết hợp với kỹ thuật xử lý song song và kiến trúc phân tán để tăng hiệu quả phát hiện các Hot-IP, phù hợp với kiến trúc mạng tổ chức dạng đa vùng.

Bài toán phát hiện các Hot-IP trực tuyến là bài toán có tính tổng quát, với các Hot-IP đại diện cho các đối tượng trên mạng hoạt động với tần suất xuất hiện cao trong một khoảng thời gian rất ngắn. Do vậy, việc ứng dụng giải pháp này cho một số bài toán an ninh mạng nhằm phát hiện sớm và hạn chế hoạt động của chúng giúp hệ thống hoạt động ổn định, thông suốt và giúp người quản trị mạng kịp thời đưa ra những biện pháp phù hợp. Một số ứng dụng từ bài toán phát hiện nhanh các Hot-IP trực tuyến như: phát hiện các đối tượng có khả năng là nguồn phát tấn công DoS/DDoS, phát hiện các đối tượng có khả năng là nạn nhân trong các cuộc tấn công DoS/DDoS, phát hiện các đối tượng có khả năng là một số loại sâu Internet dạng quét không gian địa chỉ IP để tìm kiếm các lỗ hổng để tiến hành phát tán sâu, phát hiện các đối tượng có khả năng đang hoạt động bất thường trong một hệ thống mạng cung cấp dịch vụ. Bên cạnh đó, có thể giám sát hoạt động của các Hot-IP trong một số chu kỳ thuật toán, kết hợp với giám sát tài nguyên của các thiết bị mạng để hạn chế hay ngăn chặn hoạt động của chúng nhằm đảm bảo hệ thống hoạt động ổn định, thông suốt.

1. CÁC KẾT QUẢ ĐẠT ĐƯỢC

Xuất phát từ bài toán phát hiện các đối tượng có khả năng gây nguy hại trên mạng như nguồn phát tán sâu Internet, nguồn phát động tấn công DoS/DDoS hay

nạn nhân trong các cuộc tấn công này; dựa vào địa chỉ IP trong các gói tin IP truyền qua các thiết bị định tuyến là giá trị đại diện cho các thiết bị truyền và nhận trên mạng; luận án đề xuất giải pháp phát hiện các Hot-IP trực tuyến để phát hiện các đối tượng này dựa trên phương pháp thử nhóm bất ứng biến. Các kết quả chính của luận án được tóm tắt như sau:

1) Luận án đã mô hình hóa bài toán phát hiện các Hot-IP dựa theo bài toán thử nhóm bất ứng biến và đề xuất kết hợp một số kỹ thuật để nâng cao hiệu quả của giải pháp. Trong đó, phương pháp nối mã được áp dụng vào việc phát sinh ma trận d-phân-cách tường minh để phát sinh chính xác ma trận phân cách và tối ưu không gian lưu trữ khi thực thi chương trình. Phương pháp này cho phép phát sinh từng cột của ma trận trong quá trình xử lý và tính toán. Do đó, ma trận không cần được lưu trữ toàn bộ trong khi thực thi chương trình. Để nâng cao hiệu quả của giải pháp, một số kỹ thuật được sử dụng kết hợp như lựa chọn các tham số trong thuật toán, kích thước ma trận dựa vào khả năng của vị trí triển khai; đề xuất kết hợp với kỹ thuật xử lý song song để giảm thời gian giải mã phát hiện các Hot-IP căn cứ vào tính chất của phương pháp thử nhóm bất ứng biến là các phép thử được xác định trước và độc lập nhau; đề xuất kết hợp với kiến trúc phân tán để phát hiện và cảnh báo sớm các Hot-IP trong hệ thống mạng tổ chức đa vùng, thích hợp áp dụng trong các mạng trung gian ở phía nhà cung cấp dịch vụ.

2) Luận án đã đề xuất cải tiến phương pháp thử nhóm bất ứng biến trong việc phát hiện các Hot-IP với hai thuật toán cải tiến. Thuật toán cải tiến thứ nhất *"Online Hot-IP Detecting"* cho phép tối ưu về mặt tính toán và độ chính xác khi số lượng Hot-IP thực tế cao hơn giá trị cho trước khi xây dựng ma trận bằng cách kết hợp với phương pháp *"counter-based"*. Thuật toán cải tiến thứ hai *"Online Hot-IP Preventing"* đảm bảo hệ thống mạng hoạt động ổn định, thông suốt bằng cách ngắt kết nối đối với các Hot-IP trong một chu kỳ thực hiện thuật toán.

3) Luận án đã mô hình hóa một số bài toán an ninh mạng như phát hiện các đối tượng có khả năng là nguồn phát tán sâu mạng, phát hiện đối tượng có khả năng

là các nạn nhân hay nguồn phát động tấn công trong các cuộc tấn công từ chối dịch vụ, phát hiện các thiết bị có khả năng đang hoạt động bất thường trên mạng về bài toán tìm Hot-IP trực tuyến. Bên cạnh đó, luận án cũng đề xuất giám sát các Hot-IP này kết hợp với việc theo dõi tài nguyên hệ thống để điều phối hoạt động của luồng lưu lượng chứa Hot-IP, giảm ảnh hưởng xấu đến hoạt động chung của toàn hệ thống mạng.

Các kết quả nghiên cứu và thực nghiệm cho thấy rằng giải pháp cho kết quả có độ chính xác cao, thời gian thực hiện để phát hiện các Hot-IP nhanh, có thể áp dụng triển khai vào môi trường thực tế ở phía các nhà cung cấp dịch vụ và các hệ thống mạng cung cấp dịch vụ trên môi trường Internet. Các kết quả chính của luận án được công bố ở các công trình [C1][C2][C3][C4][C5][C6][C7][C8] trong danh mục các công trình nghiên cứu của tác giả.

2. HƯỚNG PHÁT TRIỂN

Luận án đã trình bày một giải pháp hoàn chỉnh về phát hiện các Hot-IP trên mạng và một số ứng dụng trong lĩnh vực an ninh mạng. Bên cạnh việc áp dụng giải pháp vào thực tiễn, đặc biệt triển khai trên phần cứng, hướng nghiên cứu mở tiếp theo là kết hợp phân tích một số yếu tố khác trong dòng dữ liệu để nhận dạng, phân loại nguy cơ từ bài toán phát hiện các Hot-IP này.

CÁC CÔNG TRÌNH NGHIÊN CỨU CỦA TÁC GIẢ

TẠP CHÍ KHOA HỌC

[C1] **Huynh Nguyen Chinh**, Nguyen Dinh Thuc, Tan Hanh (2013). Finding Hot-IPs in network using group testing method – A review. *Journal of Engineering Technology and Education – Kuas,Taiwan*, pp.374-379.

[C2] **Huynh Nguyen Chinh**, Nguyen Dinh Thuc, Tan Hanh (2013). Group testing for detecting worms in computer networks. *Tạp chí Khoa học và Công nghệ - chuyên san các công trình nghiên cứu về Điện tử, Viễn thông và CNTT*, pp.12-19.

[C3] **Huynh Nguyen Chinh**, Tan Hanh, and Nguyen Dinh Thuc (2013). Fast detection of DDoS attacks using Non-Adaptive group testing. *International Journal of Network Security and Its Applications (IJNSA), Vol.5 (5)*, pp. 63–71, India.

[C4] **Huynh Nguyen Chinh** (2015). Fast detecting Hot-IPs in high speed networks. Tạp chí Phát Triển KH-CN, chuyên san KHTN, ĐHQG Tp.HCM, Vol 18, pp.242-253.

HỘI NGHỊ KHOA HỌC QUỐC TẾ

[C5] Thach V. Bui, **Chinh N. Huynh**, Thuc D. Nguyen (2013). Early detection for networking anomalies using Non-Adaptive Group testing. *International Conference on ICT Convergence 2013 (ICTC 2013), Korea*, pp. 984-987, IEEE.

[C6] **Huynh Nguyen Chinh**, Nguyen Dinh Thuc, Tan Hanh (2014). A distributed architecture and Non-adaptive Group testing approach to fast detect Hot-IPs in ISP networks. IEEE - 2014 *International Conference on Green and Human Information Technology (ICGHIT 2014)*, pp.232-236, IEEE.

[C7] **Huynh Nguyen Chinh**, Nguyen Dinh Thuc, Tan Hanh (2014). Early detection and limitation Hot-IPs using Non-adaptive group testing and dynamic firewall rules. *International Conference on Computing, Management and Telecommunications (ComManTel 2014),* pp. 286-290, IEEE.

[C8] **Huynh Nguyen Chinh**, Nguyen Dinh Thuc, Tan Hanh (2014). Monitoring Hot-IPs in high speed networks. *The 2014 International Conference on Advanced Technologies for Communications (ATC'14),* pp. 430-434, IEEE.

TÀI LIỆU THAM KHẢO

[1] Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *Communications Surveys & Tutorials, IEEE*, *15*(4), pp.2046-2069.

[2] Deng, Zhantao, Jin Cao, Jin He, and Sheng Li (2013). A Novel IP Traceback Scheme to Detect DDoS. In *Proceedings of the 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*. IEEE Computer Society. pp. 1077-1080.

[3] Wu, Y. C., Tseng, H. R., Yang, W., and Jan, R. H. (2011). DDoS detection and traceback with decision tree and grey relational analysis. *International. Journal of Ad Hoc and Ubiquitous Computing*, vol. 7, no. 2, pp. 121-136.

[4] Girma, Anteneh, Moses Garuba, Jiang Li, and Chunmei Liu (2015). Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment. In *Information Technology-New Generations (ITNG)*, IEEE - *2015 12th International Conference on*, pp. 212-217.

[5] Miao, Chen, Jie Yang, Weimin Li, and Zhenming Lei (2012). A DDoS Detection Mechanism Based on Flow Analysis. In *Proceedings of the 2012 International Conference on Electronics, Communications and Control*, IEEE Computer Society, pp. 2245-2249.

[6] He, Xiaowei, Shuyuan Jin, Yunxue Yang, and Huiqiang Chi (2014). DDoS Detection Based on Second-Order Features and Machine Learning. In *Trustworthy Computing and Services*, pp. 197-205. Springer Berlin Heidelberg.

[7] Nadiammai, G. V., and M. Hemalatha (2014). Effective approach toward Intrusion Detection System using data mining techniques. Egyptian Informatics Journal15, no. 1, pp. 37-50.

[8] Xylogiannopoulos, Konstantinos, Panagiotis Karampelas, and Reda Alhajj (2014). Early DDoS Detection Based on Data Mining Techniques. In *Information Security Theory and Practice. Securing the Internet of Things*, pp. 190-199. Springer Berlin Heidelberg.

[9] Prajapati, N. M., Mishra, A., & Bhanodia, P. (2014). Literature survey-IDS for DDoS attacks. In IT in Business, Industry and Government (CSIBIG), 2014 Conference on (pp. 1-3). IEEE.

[10] Cormode, G., & Hadjieleftheriou, M (2009). Finding the frequent items in streams of data. *Communications of the ACM, 52(10), pp.97-105*.

[11] Ma, Xinlei, and Yonghong Chen (2014). DDoS Detection method based on chaos analysis of network traffic entropy. *Communications Letters, IEEE* 18, no. 1 (2014), pp. 114-117.

[12] Saleh, M., & Abdul Manaf, A. (2014). Optimal specifications for a protective framework against HTTP-based DoS and DDoS attacks. InBiometrics and Security Technologies (ISBAST), 2014 International Symposium on, pp. 263-267. IEEE.

[13] Li, Y., Guo, L., Fang, B. X., Tian, Z. H and Zhang, Y. Z. (2008). Detecting DDoS Attacks Against Web Server via Lightweight TCMKNN Algorithm. *In Proc. ACM SIGCOMM*, pp.497-498.

[14] Xie, Y. and Yu, S. (2009). Monitoring the Application-Layer DDoS Attacks for Popular Websites. *IEEE Trans. on Networking*, vol. 17, No. 1. pp. 15-25.

[15] Ho, Cheng-Yuan, Yuan-Cheng Lai, I-Wei Chen, Fu-Yu Wang, and Wei-Hsuan Tai (2012). Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. *Communications Magazine, IEEE* 50, no. 3, pp. 146-154.

[16] Forney Jr. G.D (1966). *Concatenated codes*. MIT Press.

[17] David, Jisa, and Ciza Thomas (2015). DDoS Attack Detection Using Fast Entropy Approach on Flow-Based Network Traffic. *Procedia Computer Science* 50, pp. 30-36.

[18] Saied, Alan, Richard E. Overill, and Tomasz Radzik (2014). Artificial Neural Networks in the Detection of Known and Unknown DDoS Attacks: Proof-of-Concept. In *Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, pp. 309-320. Springer International Publishing.

[19] Singh, Khundrakpam Johnson, and Tanmay De (2015). DDOS Attack Detection and Mitigation Technique Based on Http Count and Verification Using CAPTCHA. In *Computational Intelligence and Networks (CINE), 2015 International Conference on*, pp. 196-197. IEEE.

[20] Zou, C. C., Towsley, D., Gong, W., & Cai, S. (2005). Routing worm: A fast, selective attack worm based on ip address information. In *Proceedings of the*

19th Workshop on Principles of Advanced and Distributed Simulation. IEEE Computer Society. pp. 199-206.

[21] Li, P., Salour, M., & Su, X. (2008). A Survey of Internet Worm Detection And Containment. *IEEE Communications Surveys and Tutorials, vol. 10, no.1, pp.20-35.*

[22] Wang, Y., Wen, S., Xiang, Y., & Zhou, W. (2014). Modeling the propagation of worms in networks: A survey. Communications Surveys & Tutorials, IEEE,16(2), pp. 942-960.

[23] Yadav, S. (2014). Target discovery schemes used by an internet worm. In Computing for Sustainable Global Development (INDIACom), 2014 International Conference on. IEEE, pp. 776-779.

[24] Kaur, R., & Singh, M. (2014). A survey on zero-day polymorphic worm detection techniques. Communications Surveys & Tutorials, IEEE, 16(3), pp. 1520-1549.

[25] Choi, Yoon-Ho, Peng Liu, and Seung-Woo Seo (2010). Creation of the importance scanning worm using information collected by Botnets. Computer Communications 33, no. 6, pp. 676-688.

[26] Simkhada, K., Taleb, T., Waizumi, Y., Jamalipour, A. & Nemoto, Y. (2009). Combating against internet worms in large-scale networks: an autonomic signature-based solution. *Security and Communication Networks, 2(1),* pp.11-28.

[27] Cormode, Graham, and S. Muthukrishnan (2005). What's hot and what's not: tracking most frequent items dynamically. *ACM Transactions on Database Systems*, Vol. 30, No. 1, pp. 249–278.

[28] Graham Cormode and S. Muthukrishnan (2005). An improved Data-stream summary: The Count-min Sketch and its Applications. *Journal of Algorithms*, vol. 55, pp.58-75.

[29] Cheraghchi, M., Hormati, A., Karbasi, A., & Vetterli, M (2011). Group testing with probabilistic tests: Theory, design and application. *Information Theory, IEEE Transactions on, 57*(10), pp. 7057-7067.

[30] Boyer, B. and Moore, J. (1982). A fast majority vote algorithm. *Technical Report 35, Institute for Computer Science, University of Texas.*

[31] Misra, J. and Gries, D. (1982). Finding repeated elements. *Science of Computer Programming, 2,* pp.*143-152.*

[32] Manku, G. and Motwani, R. (2002). Approximate frequency counts over data streams. *In Proceedings of 28th International Conference on Very Large Data Bases*, pp. *346-357.*

[33] Metwally, A., Agrawal, D., Abbadi, A.E (2005). Efficient computation of frequent and top-k elements in data streams. *In International Conference on Database Theory, pp. 398-412*

[34] Charikar, M., Chen, K. and Farach-Colton, M. (2002). Finding frequent items in data streams. *In Procedings of the International Colloquium on Automata, Languages and Programming (ICALP), pp. 693–703.*

[35] Fischer, M. and Salzberg, S. (1982). Finding a majority among n votes: Solution to problem. *Journal of Algorithms, 3(4),pp.376-379.*

[36] Graham Cormode and S. Muthukrishnan (2005). What's new: finding significant differences in network data streams. *IEEE/ACM Trans. Netw., 13(6):1219–1232.*

[37] Ying Xuan, Incheol Shin, My T. Thai, Taieb Znati. Detecting Application Denial-of-Service Attacks: A Group-Testing-Based Approach. *Parallel and Distributed Systems, IEEE Transactions on (Volume:21 , Issue: 8)*, 2010

[38] Khattab S., Bobriel S., Melhem R., and Mosse D (2008). Live Baiting for Service-level DoS Attackers. *INFOCOM.*

[39] Piotr Indyk , Hung Q. Ngo, and Atri Rudra (2010). Efficiently decodable nonadaptive group testing. In *Proceedings of the Twenty-First Annual ACMSIAM Symposium on Discrete Algorithms (SODA)*, pp. 1126-1142.

[40] Ngo, H. Q., Porat, E. and Rudra, A. (2011). Efficiently decodable error-correcting list disjunct matrices and applications. In *ICALP* (1), pp. 557–568.

[41] Cheraghchi, Mahdi (2013). Noise-resilient group testing: Limitations and constructions. *Discrete Applied Mathematics 161.1, pp.81-95*.

[42] Porat, Ely, and Amir Rothschild (2011). Explicit non-adaptive combinatorial group testing schemes. *Information Theory, IEEE Transactions on* 57.12 (2011): pp. 7982-7989.

[43] David Eppstein, Michael T. Goodrich, Daniel S. Hirschberg (2007). Improved Combinatorial Group Testing Algorithms for Real-World Problem Sizes. *SIAM J. Comput.,* 36(5), pp.1360–1375.

[44] Ali, M., Khattab, S., & Bahgat, R. (2014). Improving Detection Accuracy in Group Testing-Based Identification of Misbehaving Data Sources. In *Future*

Internet of Things and Cloud (FiCloud), 2014 International Conference on, pp. 167-174. IEEE.

[45] Kautz, W., and Roy Singleton (1964). Nonrandom binary superimposed codes. *Information Theory, IEEE Transactions on 10, No. 4, pp. 363-377.*

[46] Robert Dorfman (1943). The detection of defective members of large populations. *The Annals of Mathematical Statistics, pp. 436-440.*

[47] Chen, H.B. and Hwang, F. K. (2008). A survey on nonadaptive group testing algorithms through the angle of decoding. *J. Comb. Optim.*, 15(1), pp.49-59.

[48] Du, D. Z. and Hwang, F. K. (2006). *Pooling Designs and Non-adaptive Group Testing -Important Tools for DNA Sequencing*. World Scientific.

[49] Lo, C., Liu, M., Lynch, J. P., & Gilbert, A. C (2013). Efficient sensor fault detection using combinatorial group testing. *In Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on,* pp. 199-206.

[50] Goodrich, Michael T., and Daniel S. Hirschberg (2008). Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis. *Journal of Combinatorial Optimization*, pp. 95-121.

[51] Gregoay M. Zaverucha and Douglas R. Stinson (2009). Group testing and Batch verification. *The 4th international conference on information theoretic security, ICITS*, pp. 140-157.

[52] Goodrich, M. T., Atallah, M. J and Tamassia, R. (2005). Indexing information for data forensics. *In Third International Conference on Applied Cryptography and Network Security (ANCS)*, pp. 206-221.

[53] Du, Dingzhu, and Frank Hwang (2000). *Combinatorial group testing and its applications -2nd Edition.* World Scientific Publishing.

[54] Cheraghchi, M., Hormati, A., Karbasi, A., & Vetterli, M (2011). Group testing with probabilistic tests: Theory, design and application. *Information Theory, IEEE Transactions on, 57*(10), pp. 7057-7067.

[55] D'yachkov, A. G., Rykov, V. V. (1982). Bounds on the Length of Disjunctive Codes, Probl. Peredachi Inf., 18:3, pp. 7–13.

[56] D. Moore, C. Shannon, and J. Brown (2002). *Code Red: A case study on the spread and victims of an Inetnet worm*. In Proceeding of the Second Internet Measurement Workshop (IMW 2002).

[57] D. Moore, V. Paxon, S.Savaga, C. Shannon, S. Staniford, and Weaver (2003). *The spread of the Sapphire/Slammer worm*. Technical report, CAIDA.

[58] Wu, J., Vangala, S., Gao, L., & Kwiat, K. A. (2004). An Efficient Architecture and Algorithm for Detecting Worms with Various Scan Techniques. *Proc. Network and Distrib. Sys. Sec. Symp*.

[59] Berk, V., Bakos, G., & Morris, R (2003). Designing a Framework for Active Worm Detection on Global Networks. *Proc. 1st IEEE Int'l. Wksp. Info. Assurance*, pp. 13-23. IEEE.

[60] Callegari, Christian, Sandrine Vaton, and Michele Pagano (2008). A new statistical approach to network anomaly detection. *In Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2008), pp. 441-447.*

[61] Shon, Taeshik, Yongdae Kim, Cheolwon Lee, and Jongsub Moon (2005). A machine learning framework for network anomaly detection using SVM and GA. *In Information Assurance Workshop, 2005. IAW'05. Proceedingsfrom the Sixth Annual IEEE SMC, pp. 176-183.*

[62] Breier, J., & Branišová, J. (2015). Anomaly Detection from Log Files Using Data Mining Techniques. *In Information Science and Applications*. *Springer Berlin Heidelberg*, pp. 449-457.

[63] Ando, Shin (2007). Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. *In Data Mining, 2007. ICDM2007. Seventh IEEE International Conference on, pp. 13-22.*

[64] Wang, A., Mohaisen, A., Chang, W., & Chen, S. (2015). Delving into internet ddos attacks by botnets: Characterization and analysis. *In IEEE International Conference on Dependable Systems and Networks (DSN).*

[65] Mantur, B., Desai, A., & Nagegowda, K. S. (2015). Centralized Control Signature-Based Firewall and Statistical-Based Network Intrusion Detection System (NIDS) in Software Defined Networks (SDN). *In Emerging Research in Computing, Information, Communication and Applications*, pp. 497-506, Springer India.

[66] http://www.cisco.com/web/about/security/intelligence/network_performance_metrics.html

[67] Kenkre, P. S., Pai, A., & Colaco, L. (2015). Real time intrusion detection and prevention system. In *Proceedings of the 3rd International Conference on*

Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014, pp. 405-411. Springer International Publishing.

[68] CAIDA. URL http://www.caida.org/

[69] MAWI URL: http://mawi.ad.jp/

[70] CAIDA. URL: http://www.caida.org/data/realtime/passive/?monitor=equinix-chicago-dirA&row =timescales &col=sources &sources =src_country& graphs_sing =ts &counters_sing=packets&timescales=24

[71] MAWI. URL: http://mawi.wide.ad.jp/mawi/ditl/ditl2012/ & http://mawi.wide.ad.jp/mawi/ditl/ditl2014/

[72] WAND. URL: http://wand.net.nz/wits/ispdsl/2/

[73] WAND. URL: http://wand.net.nz/

[74] CISCO: The Zettabyte Era: Trends and Analysis – White Paper. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html (updated: June 02, 2016)

[75] Kumawat, H., & Meena, G. (2014, November). Characterization, Detection and Mitigation of Low-Rate DoS attack. *In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (p. 69). ACM.*

[76] Visoottiviseth, V. and Bureenok, N. (2008). Performance comparison of ISATAP implementations on FreeBSD, RedHat, and Windows 2003. *In Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on (pp. 547-552). IEEE*.