

ĐẠI HỌC QUỐC GIA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN

NGUYỄN ANH CHUYÊN

NGHIÊN CỨU ĐỀ XUẤT CÁC PHƯƠNG PHÁP ĐÁNH GIÁ  
ĐỘ TIN CẬY CHO CÁC CƠ CHẾ DỰ PHÒNG CỦA HỆ THỐNG MÁY CHỦ

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2024

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN**

**NGUYỄN ANH CHUYỀN**

**NGHIÊN CỨU ĐỀ XUẤT CÁC PHƯƠNG PHÁP ĐÁNH GIÁ  
ĐỘ TIN CẬY CHO CÁC CƠ CHẾ DỰ PHÒNG CỦA HỆ THỐNG MÁY CHỦ**

**Chuyên ngành: Quản lý Hệ thống thông tin**

**Mã số: 9480205.01QTD**

**LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN**

**NGƯỜI HƯỚNG DẪN KHOA HỌC**

- 1. TS. Lê Quang Minh**
- 2. PGS. TS. Nguyễn Văn Tam**

**Hà Nội – 2024**

## LỜI CAM ĐOAN

Tôi xin cam đoan luận án “*Nghiên cứu đề xuất các phương pháp đánh giá độ tin cậy cho các cơ chế dự phòng của hệ thống máy chủ*” là công trình nghiên cứu của cá nhân tôi, được hoàn thành dưới sự hướng dẫn của TS. Lê Quang Minh và PGS.TS Nguyễn Văn Tam. Các kết quả nghiên cứu của tôi cùng với các tác giả khác đã được sự nhất trí của các đồng tác giả khi đưa vào nội dung luận án. Tôi đã trích dẫn đầy đủ các tài liệu tham khảo, công trình nghiên cứu liên quan ở trong nước và quốc tế. Tôi xin cam đoan các số liệu và kết quả trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác.

*Hà Nội, ngày .. tháng .. năm 2024*

**Tác giả luận án  
Nghiên cứu sinh**

**Nguyễn Anh Chuyên**

## LỜI CẢM ƠN

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

Xin trân trọng cảm ơn Ban Giám hiệu, Ban chủ nhiệm Khoa Công nghệ thông tin trường Đại học Công nghệ Thông tin và Truyền thông - ĐHTN đã luôn tạo mọi điều kiện thuận lợi nhất cho tác giả trong suốt quá trình thực hiện luận án.

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

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

*Hà Nội, ngày .. tháng .. năm 2024*

**Tác giả luận án**

**Nguyễn Anh Chuyên**

## MỤC LỤC

LỜI CAM ĐOAN .....	i
LỜI CẢM ƠN .....	ii
MỤC LỤC .....	iii
DANH MỤC VIẾT TẮT .....	vi
DANH MỤC KÝ HIỆU .....	viii
DANH SÁCH HÌNH VẼ .....	ix
DANH SÁCH BẢNG .....	xi
PHẦN MỞ ĐẦU .....	1
CHƯƠNG 1. TỔNG QUAN VỀ ĐỘ TIN CẬY CỦA HỆ THỐNG .....	7
1.1. Tổng quan về độ tin cậy của hệ thống .....	7
1.1.1. Khái niệm độ tin cậy của phần tử và hệ thống .....	7
1.1.2. Một số thuật ngữ liên quan độ tin cậy .....	15
1.1.3. Chỉ số độ tin cậy của hệ thống .....	<b>Error! Bookmark not defined.</b>
1.1.4 Bài toán đánh giá độ tin cậy của hệ thống mạng .....	19
1.2. Tổng quan về các phương pháp tính độ tin cậy của hệ thống .....	21
1.2.1. Phương pháp liệt kê trạng thái (State Enumeration - SE) .....	21
1.2.2. Phương pháp cắt cực tiểu .....	23
1.2.3. Phương pháp tổng sản phẩm rời rạc .....	24
1.2.4. Phương pháp biểu đồ quyết định nhị phân .....	25
1.2.5. Phương pháp Simple Algorithm For Computing Network Reliability (SACNR) ..	26
1.2.6. Một số nhận xét .....	26
1.3. Tổng quan về các phương pháp đánh giá độ tin cậy của hệ thống .....	27
1.3.1. Phương pháp mô phỏng Monte Carlo .....	27
1.3.2. Phương pháp sử dụng chuỗi Markov (Markov chain) .....	29
1.3.3. Phương pháp sử dụng mạng Bayesian .....	30
1.3.4. Phương pháp sử dụng phân tích cây sai .....	31
1.3.5. Một số nhận xét .....	32
1.4. Các phương pháp dự phòng nâng cao độ tin cậy hệ thống .....	33
1.4.1. Cơ chế dự phòng nóng .....	33
1.4.2. Cơ chế dự phòng lạnh .....	36
1.4.3. Cơ chế dự phòng ấm .....	38

1.4.4. Cơ chế dự phòng kiểu chấp (dự phòng theo cơ chế bỏ phiếu) .....	39
1.4.5. Một số nhận xét.....	40
1.5. Kết luận và vấn đề nghiên cứu.....	41
<b>CHƯƠNG 2. PHƯƠNG PHÁP ĐÁNH GIÁ VÀ CẢI THIẾN TÍNH ĐỘ TIN CẬY GIỮA HAI ĐIỂM ĐẦU CUỐI TRONG MẠNG.....</b>	<b>42</b>
2.1. Vấn đề đánh giá độ tin cậy giữa hai điểm đầu cuối trong mạng.....	42
2.2. Mô hình mạng và độ tin cậy của hai thiết bị đầu cuối .....	44
2.2.1 Biểu diễn kết nối mạng trong lý thuyết đồ thị .....	44
2.2.2. Sử dụng phương thức SDP trong tính xác suất tổng các thành phần.....	46
2.3. Phương pháp tính toán độ tin cậy hai nút đầu cuối sử dụng thuật toán PNRE (Parallel Network Reliability Evaluation).....	48
2.3.1 Tính toán xác suất của biểu thức logic dựa trên LPC .....	48
2.3.2 Lưu đồ hoạt động của thuật toán PNRE .....	49
2.3.3. Đánh giá độ phức tạp thuật toán .....	51
2.4 Cài đặt thuật toán PNRE.....	52
2.4.1 Xác định trực giao hoá các toán tử logic .....	52
2.4.2 Một số giải thuật được cài đặt trong thuật toán PNRE .....	56
2.5 Thực nghiệm và so sánh phương pháp PNRE với LPC, SACNR.....	61
2.6. Kết luận chương.....	66
<b>CHƯƠNG 3. QUY TRÌNH ĐẢM BẢO ĐỘ TIN CẬY CHO HỆ THỐNG MÁY CHỦ DỰA TRÊN CƠ CHẾ DỰ PHÒNG.....</b>	<b>68</b>
3.1. Cơ chế dự phòng nâng cao độ tin cậy cho hệ thống.....	68
3.1.1. Phương pháp dự phòng song song.....	69
3.1.2. Dự phòng song song với phần tử có phục hồi .....	70
3.1.3. Phương pháp dự phòng tích cực .....	71
3.2. Bài toán đảm bảo độ tin cậy cho hệ thống máy chủ .....	72
3.2.1. Phát biểu nội dung bài toán.....	72
3.2.2. Đề xuất quy trình đảm bảo độ tin cậy cho hệ thống. ....	74
3.3. Nâng cao độ tin cậy sử dụng phương pháp dự phòng song song .....	77
3.3.1 Bài toán nâng cao độ tin cậy cho hệ thống máy chủ dịch vụ.....	77
3.3.2 Đảm bảo độ tin cậy hệ thống sử dụng dự phòng song song .....	92
3.4. Nâng cao độ tin cậy sử dụng phương pháp dự phòng tích cực.....	101
3.4.1 Tính độ tin cậy hệ thống với dự phòng tích cực .....	101

3.4.2 Bài toán lưu trữ dữ liệu an toàn .....	102
3.4.3 Đảm bảo độ tin cậy hệ thống với cơ chế dự phòng tích cực.....	106
3.5. Tổng kết chương.....	112
KẾT LUẬN.....	114
DANH MỤC CÔNG TRÌNH TÁC GIẢ ĐÃ CÔNG BỐ .....	116
TÀI LIỆU THAM KHẢO .....	118

## DANH MỤC VIẾT TẮT

<b>Viết tắt</b>	<b>Tiếng Anh</b>	<b>Tiếng Việt</b>
AP	Active Protection	Phương pháp dự phòng chủ động
BDD	Binary Decision Diagram	Lược đồ quyết định nhị phân
BN	Bayesian Network	Mạng Bayes
CA	Cellular Automata	Automat dạng lưới ô nhỏ (cell)
CCF	Common Cause Failures	Các nguyên nhân lỗi phổ biến
CDNs	Content Delivery Networks	Các mạng phân phối nội dung
CPU	Central Processing Unit	Khối xử lý trung tâm
DAG	Directed Acyclic Graph	Đồ thị tuần hoàn có hướng
FTA	Fault Tree Analysis	Phân tích cây lỗi
FUSA	Fixed-Node Unconnected Subgraphs Algorithm	Thuật toán đồ thị con không được kết nối với số nút cố định
GPU	Graphics Processing Unit	Khối xử lý đồ họa
IC	Integrated Circuit	Mạch tích hợp
IoT	Internet of things	Internet vạn vật
IP	Internet Protocol	Giao thức Internet
IPC	Incomplete Coverage	Độ phủ không đầy đủ
LPC	Logical-Probabilistic Calculus	Phép tính logic xác suất
MC	Minimal Cut	Cắt cực tiểu
MCS	Monte Carlo Simulation	Mô phỏng Monte Carlo
MTBF	Mean Time Between Failure	Thời gian trung bình giữa hai lần hỏng
MTTF	Mean Time To Failure	Thời gian hoạt động an toàn trung bình
MTTR	Mean Time To Repair	Thời gian trung bình sửa chữa sự cố
NP-Hard	Nondeterministic Polynomial	Độ khó đa thức không xác định
ODNF	Orthogonal Disjunctive Normal Form	Dạng chuẩn tắc trực giao
PDS	Primary DNS Server	Máy chủ DNS chính

PNRE	Parallel Network Reliability Evaluation	Đánh giá độ tin cậy của mạng song song
QFUS	Quick Fixed-Node Unconnected Subgraphs Algorithm	Thuật toán đồ thị con không được kết nối với nút cố định nhanh
RAID	Redundant Arrays of Independent Disks	Mảng dự phòng các ổ đĩa độc lập
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
RESCS	RAID Enhance Security Cloud Storage	Nâng cao bảo mật lưu trữ đám mây dựa trên RAID
SACNR	Simple Algorithm For Computing Network Reliability	Thuật toán đơn giản để tính độ tin cậy mạng
SDP	Sum-of-Disjoint Products	Tổng các thành phần rời rạc
SDS	Secondary DNS Server	Máy chủ DNS phụ
SE	State Enumeration	Liệt kê trạng thái
TCP	Transmission Control Protocol	Giao thức điều khiển đường truyền
UAV	Unmanned Aerial Vehicle	Thiết bị bay không người lái
UDP	User Datagram Protocol	Giao thức gói dữ liệu người dùng
USB	Universal Serial Bus	Bus tuần tự đa năng
WSN	Wireless Sensor Network	Mạng cảm biến không dây

## DANH MỤC KÝ HIỆU

$\tau$	Thời gian vận hành liên tục một cách an toàn của phần tử
$\Delta t$	Khoảng thời gian rất nhỏ dùng xét hoạt động của phần tử
$P(t)$	Xác suất hoạt động ổn định đến thời điểm $t$ của phần tử.
$Q(t)$	Xác suất xảy ra lỗi tại thời điểm $t$ của phần tử
$P(A)$	Xác suất xảy ra sự kiện $A$
$q(t)$	Hàm mật độ phân phối xác suất
$\lambda(t)$	Cường độ hỏng hóc của phần tử
$R$	Độ tin cậy của phần tử trong quá trình làm việc.
$\rho$	Biểu thức tương quan giữa xác suất hỏng và phục hồi.
$\lambda$	Xác suất hỏng của phần tử
$\mu$	Xác suất phục hồi của phần tử
$\alpha, \beta, \gamma$	Tương ứng là độ tin cậy của từng khối lưu trữ dữ liệu.
$e^{-\lambda t}$	Độ tin cậy của phần tử theo thời gian $t$
$K_i$	Mệnh đề sơ cấp có số thứ tự là $i$ trong hàm logic.

## DANH SÁCH HÌNH VẼ

Hình 1.1: Biểu diễn hàm mật độ phân phối xác suất .....	9
Hình 1.2: Biểu diễn sự biến đổi xác suất hỏng theo thời gian của phần tử .....	11
Hình 1.3: Biểu diễn độ tin cậy của phần tử theo thời gian .....	11
Hình 1.4: Tương quan giữa độ tin cậy và độ hỏng của phần tử.....	14
Hình 1.5: Biểu diễn cường độ hỏng của phần tử .....	14
Hình 1.6: Mối liên hệ giữa các thống số MTBF, MTTR và MTTF. ....	17
Hình 1.7: Mối liên hệ giữa MTTF và MTBF [67].....	18
Hình 1.8: Hệ thống các phần tử trong dự phòng nóng.....	35
Hình 1.9: Cơ chế dự phòng nóng của tổng đài chăm sóc khách hàng.....	36
Hình 1.10: Cơ chế dự phòng lạnh cho hệ thống máy chủ.....	38
Hình 1.11: Phương pháp dự phòng theo cơ chế chập ba .....	39
Hình 2.1: Đồ thị với đỉnh nguồn “0” và đích “1” .....	46
Hình 2.2: Lưu đồ thuật toán PNRE.....	50
Hình 2.3: Tiến hành tính toán song song hóa các biểu thức trong Công thức (2.11) .....	51
Hình 3.1: Sơ đồ chuyển trạng thái Markov với các phần tử phục hồi .....	70
Hình 3.2: Cơ chế dự phòng tích cực .....	72
Hình 3.3: Quy trình đảm bảo độ tin cậy cho hệ thống.....	77
Hình 3.4: Mô hình hoạt động của hệ thống máy chủ DNS Anycast.....	78
Hình 3.5: Mô hình hai máy chủ DNS Anycast hoạt động song song .....	79
Hình 3.6: Sơ đồ chuyển trạng thái của hệ thống với hai phần tử.....	80
Hình 3.7: Mô hình ba máy chủ DNS hoạt động có dự phòng .....	82
Hình 3.8: Sơ đồ chuyển trạng thái của hệ với ba phần tử song song.....	83
Hình 3.9: Sơ đồ chuyển trạng thái với hai phần tử phục hồi có ưu tiên khác nhau .....	86
Hình 3.10: Tương quan hệ số sẵn sàng trong hai trường hợp.....	89
Hình 3.11: Biểu đồ so sánh các phương án dự phòng với phần tử không phục hồi. ....	90
Hình 3.12: Hệ thống ban đầu không có dự phòng. ....	93
Hình 3.13: Các phương án dự phòng khả thi của hệ thống. ....	95
Hình 3.14: Thông số về tỉ lệ hỏng của thiết bị máy phát. ....	98
Hình 3.15: So sánh độ tin cậy của các phương án dự phòng từ Bảng 3.3. ....	100
Hình 3.16: Mô hình hoạt động của cơ chế lưu trữ RESCS.....	104
Hình 3.17: Trường hợp sử dụng hai phần tử dự phòng tích cực.....	105

Hình 3.18: Trường hợp sử dụng ba phần tử dự phòng .....	105
Hình 3.19: Các phương án dự phòng tích cực của hệ thống.....	108
Hình 3.20: So sánh độ tin cậy giữa các phương án dự phòng tích cực.....	111

## DANH SÁCH BẢNG

### No table of figures entries found.

Bảng 1.1: Một số thông số về độ tin cậy thường được sử dụng .....	15
Bảng 2.1: Các đường dẫn tối thiểu từ $s$ đến $t$ trong Hình 2.1 .....	47
Bảng 2.2: So sánh thuật toán PNRE và phương pháp SDP .....	51
Bảng 2.3: Các mô hình (topo) mạng được sử dụng trong thực nghiệm.....	61
Bảng 2.4: Thông số các mạng được dùng trong thực nghiệm .....	64
Bảng 2.5: So sánh thời gian thực hiện của PNRE với các thuật toán khác .....	65
Bảng 3.1: So sánh hệ số sẵn sàng giữa hai trường hợp phân tử hệ thống.....	88
Bảng 3.2: So sánh độ tin cậy của hệ thống sau các mốc thời gian .....	91
Bảng 3.3: Giá trị độ tin cậy của các phương án sử dụng dự phòng song song.....	98
Bảng 3.4: Giá trị độ tin cậy của các phương án sử dụng dự phòng tích cực .....	110

## PHẦN MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong bối cảnh hiện nay, hệ thống máy tính và mạng đóng vai trò vô cùng quan trọng trong hầu hết các lĩnh vực của cuộc sống, từ công nghiệp, thương mại, giáo dục đến y tế và dịch vụ công. Sự phụ thuộc ngày càng lớn vào các hệ thống này đặt ra yêu cầu ngày càng cao hơn về độ tin cậy và khả năng hoạt động liên tục, đặc biệt khi một sự cố nhỏ cũng có thể gây ra những hậu quả nghiêm trọng. Để đáp ứng nhu cầu này, việc nghiên cứu và áp dụng các phương pháp nhằm mục đích nâng cao độ tin cậy cho hệ thống trở nên cần thiết hơn bao giờ hết.

Hiện nay, có nhiều phương pháp để nâng cao độ tin cậy cho hệ thống như: tăng cường vai trò của kiểm thử hệ thống; thực hiện giám sát hoạt động của hệ thống và cảnh báo khi sự cố; thực hiện bảo trì hệ thống định kỳ, thường xuyên; nâng cao phương pháp bảo mật cho hệ thống... Tuy nhiên, phương pháp dự phòng là một trong những giải pháp được đánh giá là hiệu quả giúp tăng cường độ tin cậy của hệ thống. Bằng cách triển khai các thành phần dự phòng, hệ thống có thể tự động chuyển đổi và khắc phục khi xảy ra lỗi, đảm bảo tính liên tục và an toàn cho quá trình vận hành. Tuy nhiên, không phải phương pháp dự phòng nào cũng mang lại hiệu quả như nhau, và việc đánh giá, so sánh để lựa chọn phương pháp phù hợp là một thách thức lớn.

Bên cạnh đó, vấn đề độ tin cậy trong mạng, đặc biệt là độ tin cậy giữa hai điểm đầu cuối trong mạng, cũng là một lĩnh vực đáng quan tâm. Sự ổn định và an toàn trong việc truyền tải dữ liệu giữa các điểm đầu cuối không chỉ đảm bảo hiệu suất hoạt động của mạng mà còn bảo vệ dữ liệu khỏi mất mát hoặc hỏng hóc. Đặc biệt, trước nguy cơ ngày càng gia tăng từ các cuộc tấn công mạng, việc nâng cao

độ tin cậy và khả năng sẵn sàng của hệ thống trở thành yếu tố then chốt để đảm bảo an ninh mạng và bảo vệ thông tin quan trọng.

Ngoài ra, việc đảm bảo độ tin cậy cho hệ thống dựa trên các yêu cầu đặt ra từ ban đầu khi xây dựng hệ thống chưa có một phương pháp, quy trình hay cách thức nào cụ thể. Điều này tạo ra một khoảng trống lớn trong việc đảm bảo độ tin cậy và an toàn của các hệ thống hiện đại. Sự thiếu hụt này không chỉ làm giảm hiệu quả vận hành mà còn tăng nguy cơ đối với an ninh và an toàn thông tin.

Xuất phát từ những lý do trên, việc nghiên cứu các giải pháp dự phòng mới nhằm nâng cao độ tin cậy và đảm bảo hoạt động của hệ thống là vấn đề cấp thiết. Đề tài “*Nghiên cứu đề xuất các phương pháp đánh giá độ tin cậy cho các cơ chế dự phòng của hệ thống máy chủ*” được thực hiện trong khuôn khổ luận án tiến sỹ chuyên ngành Quản lý hệ thống thông tin. Đề tài này không chỉ mang lại những hiểu biết sâu sắc về các phương pháp dự phòng khác nhau mà còn cung cấp những giải pháp cụ thể để cải thiện độ tin cậy của hệ thống mạng, góp phần nâng cao hiệu quả và an toàn trong quá trình vận hành. Việc mở rộng phạm vi nghiên cứu bao gồm cả độ tin cậy trong mạng, khả năng sẵn sàng trước các cuộc tấn công mạng, và xây dựng phương pháp đảm bảo độ tin cậy từ giai đoạn thiết kế sẽ làm cho đề tài trở nên hấp dẫn và có tính ứng dụng cao hơn.

Nội dung luận án trình bày các kết quả nghiên cứu liên quan tới: Đánh giá các yếu tố ảnh hưởng đến độ tin cậy của hệ thống, cũng như các giải pháp và quy trình để nâng cao độ tin cậy, tính sẵn sàng; Thực hiện phân tích và đề xuất phương án dự phòng của hệ thống máy chủ trong môi trường điện toán đám mây; Cải thiện tốc độ tính toán và hiệu suất làm việc của phương pháp tính độ tin cậy trong mạng; Xây dựng quy trình đảm bảo độ tin cậy của hệ thống trước khi triển khai.

## **2. Mục đích nghiên cứu**

Mục đích nghiên cứu chính của luận án là các phương pháp dự phòng nâng cao độ tin cậy cho hệ thống nhằm đảm bảo tính ổn định, sẵn sàng trong quá trình hoạt động.

Nội dung nghiên cứu của luận án tập trung vào các chủ điểm sau đây:

- **Mục tiêu thứ nhất:** Nghiên cứu và cải thiện hiệu quả tính độ tin cậy giữa hai thiết bị đầu cuối trong mạng thông qua cơ chế song song hóa các tác vụ tính toán.
- **Mục tiêu thứ hai:** Nghiên cứu và đánh giá các cơ chế dự phòng, từ đó đề xuất quy trình đảm bảo độ tin cậy theo cấu trúc hệ thống dựa trên các phương pháp dự phòng song song và dự phòng tích cực.

### **3. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu: Các phương pháp tính toán độ tin cậy và dự phòng nâng cao độ tin cậy cho hệ thống.

Phạm vi nghiên cứu: Các phương pháp dự phòng nâng cao độ tin cậy cho hệ thống và một số kỹ thuật tính độ tin cậy cho hệ thống dựa trên tính toán xác suất và lý thuyết độ tin cậy.

### **4. Nội dung nghiên cứu**

Luận án tập trung nghiên cứu các nội dung:

- Tổng quan về độ tin cậy của hệ thống, các phương pháp được sử dụng để tính độ tin cậy hệ thống, phương pháp đánh giá và dự phòng nâng cao độ tin cậy hệ thống được sử dụng phổ biến hiện nay.

- Nghiên cứu và đề xuất cải thiện phương pháp tính độ tin cậy giữa hai điểm đầu cuối trong mạng dựa trên phương pháp truyền thống SDP, kết hợp với kỹ thuật tính toán song song để cải thiện hiệu quả tính toán của phương pháp.

- Nghiên cứu và đề xuất quy trình đảm bảo độ tin cậy của hệ thống dựa trên cơ chế dự phòng, thực hiện đánh giá và so sánh các phương án dự phòng song song, dự phòng tích cực dựa trên cấu trúc của hệ thống.

## **5. Phương pháp nghiên cứu**

- Phương pháp tổng kê tổng hợp: Phương pháp này được sử dụng để thu thập, tổng hợp các tài liệu kỹ thuật, các công bố khoa học các tài liệu liên quan đến đảm bảo độ tin cậy, các phương pháp dự phòng, kỹ thuật tính độ tin cậy, cơ sở lý thuyết về xác suất thống kê, mô hình toán học về chuỗi Markov. Từ đó, luận án phân tích và lựa chọn hướng tiếp cận khoa học, phù hợp.

- Nghiên cứu lý thuyết: Phương pháp này được sử dụng để tìm hiểu về các tài liệu kỹ thuật, các công trình nghiên cứu, các bài báo khoa học về các cơ chế dự phòng đảm bảo độ tin cậy trong hệ thống, phương pháp tính độ tin cậy trong mô hình mạng. Từ đó, luận án tiến hành phân tích, tổng hợp và đưa ra các vấn đề cần nghiên cứu.

- Phương pháp chuyên gia: Phương pháp nghiên cứu chuyên gia được sử dụng bằng cách tham gia các hội thảo khoa học nhằm trao đổi các kinh nghiệm, thu thập các ý kiến đóng góp của các chuyên gia và tích cực trao đổi với các chuyên gia nước ngoài.

- Phương pháp kiểm chứng: Áp dụng cơ sở lý thuyết về độ tin cậy để xây dựng, xác lập công thức tính độ tin cậy dựa trên các phương pháp dự phòng cụ thể được áp dụng, từ đó tính toán, phân tích và đưa ra đề xuất giải pháp.

## **6. Đóng góp của luận án**

1- Đã đề xuất phương pháp PNRE nhằm cải tiến thuật toán truyền thống SDP để tính độ tin cậy giữa hai thiết bị đầu cuối trong hệ thống mạng. Bằng cách thực hiện song song hóa các hàm tính độ tin cậy của mỗi thành phần con trong

đường đi từ điểm nguồn đến đích, phương pháp đã cho kết quả tính toán được cải thiện đáng kể so sánh với hai thuật toán cùng loại là LPC và SACNR.

2- Nghiên cứu và đánh giá hiệu quả của một số phương án dự phòng nâng cao độ tin cậy cho hệ thống theo phương pháp dự phòng song song và dự phòng tích cực. Từ đó đề xuất quy trình đảm bảo độ tin cậy cho hệ thống dựa theo cấu trúc nhằm xác định phương án triển khai hệ thống hoạt động đảm bảo độ tin cậy, sẵn sàng.

## **7. Bố cục luận án**

Luận án gồm các phần sau:

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

*Chương 1* trình bày tổng quan về độ tin cậy của hệ thống, các khái niệm liên quan tới độ tin cậy, các phương pháp phổ biến được sử dụng để tính độ tin cậy trong hệ thống, các phương pháp đánh giá độ tin cậy và cơ chế dự phòng được sử dụng trong việc nâng cao độ tin cậy hệ thống. Thông qua các lập luận, phân tích để rút ra vấn đề mấu chốt cần tập trung giải quyết liên quan tới độ tin cậy hệ thống, đồng thời xây dựng các luận điểm chính của luận án và làm cơ sở nghiên cứu các vấn đề liên quan trong các chương tiếp theo.

*Chương 2* trình bày các nghiên cứu về đánh giá và nâng cao hiệu quả tính độ tin cậy giữa hai điểm đầu cuối trong mạng. Ngoài việc trình bày các cơ sở liên quan tới lý thuyết đồ thị để biểu diễn mô hình mạng qua các nút và cạnh, kỹ thuật SDP để tính độ tin cậy giữa hai điểm đầu cuối, chương này trình bày các đề xuất nhằm cải tiến phương pháp tính của SDP bằng việc song song hóa quá trình thực hiện các thao tác liên quan tới trực giao hóa ma trận. Kết quả cài đặt và thử nghiệm

phương pháp mới PNRE được so sánh với một số phương pháp truyền thống có liên quan trên một số mô hình mạng, dựa trên kết quả để đánh giá tính vượt trội và cải tiến của PNRE.

*Chương 3* trình bày các cơ chế dự phòng nâng cao độ tin cậy hệ thống như phương pháp dự phòng song song, dự phòng tích cực, từ đó đánh giá và thử nghiệm trên một số cấu hình dự phòng để so sánh mức độ cải thiện cụ thể. Bên cạnh đó, nội dung chương cũng đưa ra đề xuất quy trình đảm bảo độ tin cậy cho hệ thống dựa trên các phương pháp dự phòng, sau đó tiến hành thử nghiệm, áp dụng quy trình để tìm ra phương án tối ưu nhất đáp ứng yêu cầu về độ tin cậy của hệ thống dựa trên các tiêu chí kỹ thuật ban đầu.

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

## CHƯƠNG 1. TỔNG QUAN VỀ ĐỘ TIN CẬY CỦA HỆ THỐNG

Nội dung chương này trình bày về một số vấn đề liên quan tới độ tin cậy và tính sẵn sàng của hệ thống. Một số kiến thức cơ sở liên quan đến độ tin cậy, mô hình đảm bảo độ tin cậy, các phương pháp phổ biến được sử dụng để tính độ tin cậy hệ thống, các phương pháp đánh giá độ tin cậy và một số cơ chế dự phòng nâng cao độ tin cậy của hệ thống. Dựa trên nội dung tổng hợp được để đưa ra các vấn đề còn tồn tại và hướng nghiên cứu chính của luận án.

### 1.1. Tổng quan về độ tin cậy của hệ thống

#### 1.1.1. Khái niệm độ tin cậy của phần tử và hệ thống

##### a. Khái niệm về hệ thống, phần tử

Trong lĩnh vực nghiên cứu về máy tính, hai khái niệm phần tử và hệ thống được sử dụng rất thường xuyên.

*“Hệ thống là một tập hợp gồm nhiều phần tử tương tác, có các mối quan hệ ràng buộc lẫn nhau và cùng hoạt động hướng tới một mục tiêu chung thông qua chấp thuận các đầu vào, biến đổi có tổ chức để tạo kết quả đầu ra”.*

Hay *“Hệ thống là một tập hợp gồm nhiều phần tử có các mối quan hệ ràng buộc tương tác lẫn nhau để thực hiện một mục đích chung”.*

Khái niệm này có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ kỹ thuật, kinh tế, xã hội học đến sinh học. Trong kỹ thuật, hệ thống máy tính bao gồm: phần cứng (CPU, RAM, ổ cứng), phần mềm (hệ điều hành, ứng dụng) và các thiết bị ngoại vi (chuột, bàn phím). Hay trong hệ thống điện bao gồm các thành phần như nhà máy phát điện, đường dây truyền tải và hệ thống phân phối điện.

Trong khi đó, phần tử được định nghĩa như là một bộ phận tạo thành hệ thống mà trong quá trình nghiên cứu độ tin cậy nhất định nó được xem như là một tổng thể không chia cắt được (ví dụ như: linh kiện, thiết bị...).

*b. Định nghĩa về độ tin cậy*

Độ tin cậy (Reliability) là một khái niệm quan trọng trong nhiều lĩnh vực, thể hiện khả năng của một hệ thống, sản phẩm hoặc dịch vụ hoạt động ổn định và không gặp lỗi trong một khoảng thời gian xác định hoặc dưới các điều kiện nhất định.

Theo đó, độ tin cậy được định nghĩa như sau: *Độ tin cậy  $P(t)$  của phần tử hoặc của hệ thống là xác suất để trong suốt khoảng thời gian khảo sát  $t$ , phần tử đó hoặc hệ thống đó vận hành an toàn trong các điều kiện hoạt động cụ thể [89].*

Theo định nghĩa ở trên, thì hệ thống được tạo nên bởi nhiều phần tử, và một phần tử cũng có thể coi là một hệ thống đơn giản. Chính vì vậy, độ tin cậy của Hệ thống được tính toán và đánh giá bởi độ tin cậy của từng thành phần, từng phần tử tạo nên hệ thống thông qua cấu trúc logic, kết nối vật lý của hệ thống.

Độ tin cậy  $P(t)$  được định nghĩa bởi biểu thức:

$$P(t) = P\{\tau \geq t\} \quad (1.1)$$

Trong đó:  $\tau$  là thời gian vận hành liên tục một cách an toàn của phần tử.

Dựa vào Biểu thức (1.1), có thể hiểu rằng: Phần tử muốn vận hành an toàn trong khoảng thời gian  $t$  thì giá trị của  $t$  phải nhỏ hơn giá trị quy định  $\tau$ .

Đồng thời, biểu thức cũng thể hiện rằng phần tử chỉ hoạt động an toàn với một xác suất nằm trong khoảng từ 0 đến 1 ( $0 \leq P \leq 1$ ) trong thời gian  $t$ . Ở thời điểm ban đầu, khi đảm bảo các yêu cầu về mặt kỹ thuật sau khi được sản xuất, trạng thái của phần tử luôn hoạt động tốt khi  $t = 0$ , tức  $P(0) = 1$ . Trong quá trình hoạt động tiếp theo của phần tử, khả năng hoạt động an toàn của phần tử sẽ suy giảm, hay xác suất để phần tử xảy ra hỏng sẽ xuất hiện với tỉ lệ cao dần. Quá trình diễn ra tới khi  $t \rightarrow \infty$  theo quy luật phát triển của vật chất trong quá trình tác động tàn phá của thời gian, phần tử sẽ chuyển sang trạng thái ngừng hoạt động hoàn

toàn, tức là xác suất hoạt động  $P(\infty) = 0$ . Thời gian hoạt động của phần tử hay còn gọi là tuổi thọ là khác nhau và phụ thuộc vào nhiều yếu tố. Với các thiết bị, linh kiện điện tử, nghiên cứu chỉ ra rằng: thời gian hoạt động ổn định của các thiết bị là trong phạm vi 5 năm [53].

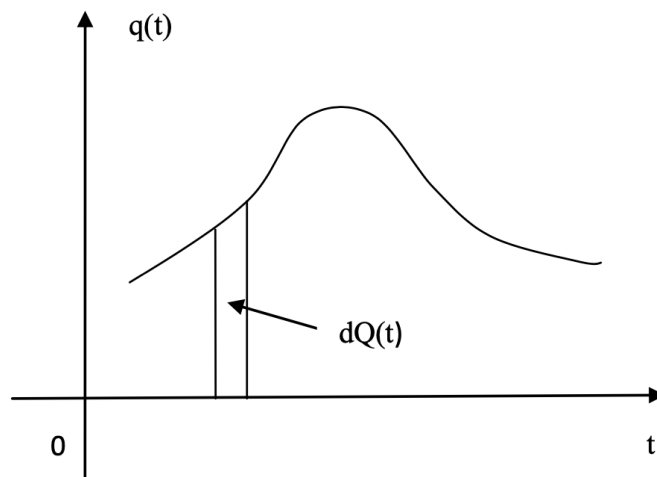
c. *Mối liên hệ giữa độ tin cậy và xác suất hỏng của phần tử*

Giả sử tại thời điểm  $t = 0$  phần tử bắt đầu hoạt động và đến thời điểm  $t = \tau$  thì phần tử gặp sự cố. Khoảng thời gian đó được gọi là thời gian vận hành an toàn một cách liên tục của phần tử. Vì sự cố không xảy ra tất định nên  $\tau$  là một đại lượng ngẫu nhiên có các giá trị trong khoảng  $0 \leq \tau \leq \infty$ .

Giả thiết trong khoảng thời gian khảo sát  $t$  thì phần tử xảy ra sự cố với xác suất  $Q(t)$ . Khi đó:  $Q(t) = P\{\tau < t\}$  (1.2)

Do  $\tau$  là đại lượng ngẫu nhiên liên tục nên:

- $Q(t)$  được gọi là hàm phân phối của biến ngẫu nhiên liên tục  $\tau$ .
- $q(t)$  là hàm mật độ phân phối xác suất của  $\tau$ .



Hình 1.1: Biểu diễn hàm mật độ phân phối xác suất

Trên Hình 1.1, biểu diễn hàm mật độ phân phối xác suất của thời gian trung bình vận hành an toàn. Theo tính chất của hàm mật độ phân phối xác suất của biến ngẫu nhiên liên tục, ta có:

$q(t) = Q'(t)$  {Đạo hàm bậc nhất của hàm phân phối xác suất}, do đó:

$$q(t) = \frac{dQ(t)}{dt} \quad (1.3)$$

Thỏa mãn tính chất là:

$$\int_0^{\infty} q(t) \cdot dt = 1$$

Vậy hàm mật độ phân phối xác suất của là:

$$q(t) = \frac{1}{\Delta t} P(t < \tau \leq t + \Delta t) \quad (1.4)$$

Trong đó:  $q(t) \cdot \Delta t$  là xác suất để thời gian hoạt động  $\tau$  nằm trong khoảng  $(t \rightarrow t + \Delta t)$  với  $\Delta t$  đủ nhỏ.

Ta có hàm  $Q(t)$  mô tả xác suất xảy ra sự cố của phần tử, vậy hàm mô tả độ tin cậy của phần tử được ký hiệu là  $P(t)$  và sẽ được tính theo định nghĩa hàm xác suất:

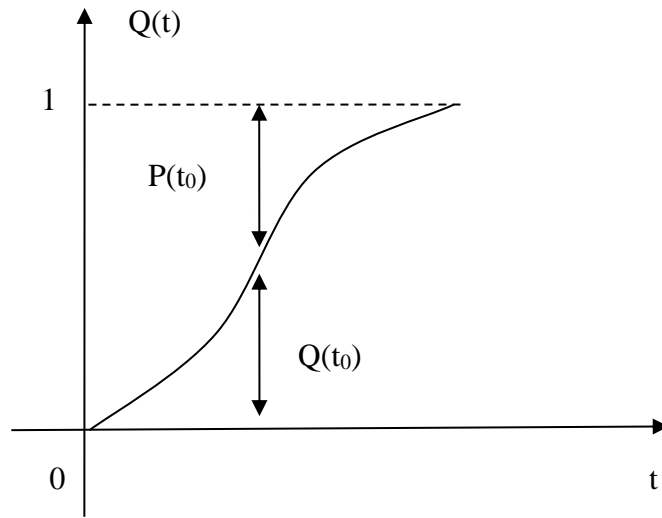
$$P(t) = 1 - Q(t) = P\{\tau \geq t\} \quad (1.5)$$

Như vậy  $P(t)$  là xác suất để phần tử vận hành an toàn trong khoảng thời gian  $t$  vì ở đây ta đã giả thiết có  $\tau \geq t$ .

$$\text{Từ biểu thức (1.3) ta có: } Q(t) = \int_0^t q(t) \cdot dt \quad (1.6)$$

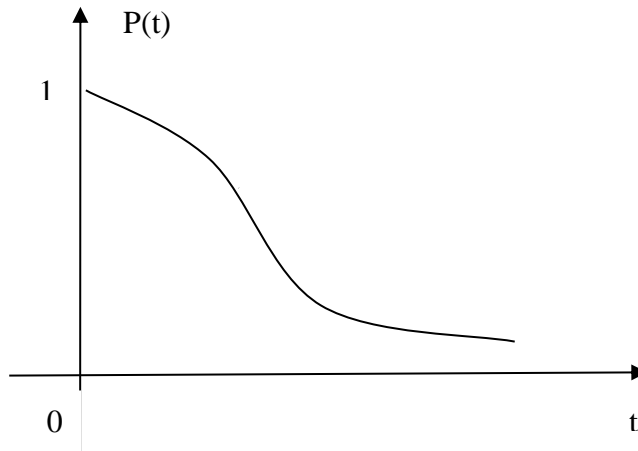
$$\text{Từ biểu thức hai biểu thức (1.5) và (1.6) ta có: } P(t) = \int_t^{\infty} q(t) \cdot dt \quad (1.7)$$

Biểu diễn hàm phân phối xác suất:



Hình 1.2: Biểu diễn sự biến đổi xác suất hỏng theo thời gian của phần tử

Hình 1.2 cho thấy sự phụ thuộc của độ hỏng vào thời gian, khi thời gian sử dụng càng lớn, xác suất xuất hiện hỏng của phần tử càng tăng lên.



Hình 1.3: Biểu diễn độ tin cậy của phần tử theo thời gian

Từ hai đồ thị trong Hình 1.2 và 1.3, ta thấy rằng  $Q(\infty) = 1$  và  $P(\infty) = 0$ , cho thấy sau một thời gian hoạt động thì độ tin cậy của phần tử giảm dần, ngược lại, xác suất hỏng của phần tử sẽ tăng dần theo thời gian.

Trong thực tế có thể thấy rằng, các thiết bị điện tử sau thời gian làm việc sẽ trở nên kém hiệu quả và dễ xảy ra trục trặc, điều này dẫn đến độ tin cậy của phần

tử giảm xuống. Theo tài liệu kỹ thuật công bố của Cơ quan Hàng không và Vũ trụ Quốc gia của chính phủ Hoa Kỳ, các linh kiện điện tử có thời gian hoạt động tốt trong khoảng phạm vi 5 năm đầu tiên [53], sau thời gian đó, tốc độ hỏng của thiết bị sẽ tăng cao và cần được thay thế để đảm bảo hoạt động của cả hệ thống.

*d. Cường độ hỏng của phần tử -  $\lambda(t)$*

Trong lý thuyết về độ tin cậy, giá trị cường độ hỏng hóc (hay cường độ trở ngại) là một trong những khái niệm quan trọng, trong đó  $\lambda(t)$  là một hàm theo thời gian [64, 89]. Với giá trị  $\Delta t$  đủ nhỏ thì biểu thức  $\lambda(t).\Delta t$  chính là xác suất để phần tử đã hoạt động tốt đến thời điểm  $t$  sẽ hỏng hóc trong khoảng thời gian  $\Delta t$  tiếp theo. Hay đó chính là số lần hỏng hóc trên một đơn vị thời gian trong khoảng thời gian  $\Delta t$ .

Cường độ hỏng được tính theo công thức:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} P(t < \tau < t + \Delta t | \tau > t) \quad (1.8)$$

Trong đó  $P(\tau > t)$  là xác suất có điều kiện, hay xác suất để phần tử hỏng hóc trong khoảng thời gian từ  $t$  đến  $(t + \Delta t)$  (sự kiện A) nếu phần tử đó hoạt động tốt đến thời điểm  $t$  (sự kiện B);  $\tau$  là thời gian vận hành an toàn.

Theo lý thuyết xác suất, xác suất nhân giữa hai sự kiện A và B là:

$$P(AB) = P(A).P(B|A) = P(B).P(A|B)$$

Hay:

$$P(A|B) = \frac{P(AB)}{P(B)}$$

Nếu  $A \subset B$  (Nếu A xảy ra thì B xảy ra) theo giả thiết ban đầu khi  $\Delta t \rightarrow 0$  thì ta có:  $P(AB) = P(A)$

Và công thức:

$$P(t < \tau < t + \Delta t | \tau > t) = \frac{P(t < \tau < t + \Delta t)}{P(\tau > t)} \quad (1.9)$$

Từ (1.8) và (1.9) ta có được:

$$\lambda(t) = \frac{q(t)}{P(t)} = \frac{q(t)}{1 - Q(t)} \quad (1.10)$$

Công thức (1.10) cho thấy mối quan hệ giữa bốn đại lượng: cường độ hỏng, hàm mật độ xác suất, hàm phân bố xác suất và độ tin cậy của phần tử.

Vậy độ tin cậy của phần tử được tính như sau:

Từ (1.3) và (1.5) ta có:

$$q(t) = Q'(t) = (1 - P(t))' = -P'(t) = -\frac{dP(t)}{dt} \quad (1.11)$$

Thay vào (1.10) ta có:

$$\begin{aligned} \lambda(t) &= -\frac{dP(t)}{P(t)dt} \\ \Rightarrow -\int_0^t \lambda(t)dt &= \int_0^t \frac{dP(t)}{P(t)} = \ln P(t) \end{aligned}$$

Do  $P(0) = 1$ :

$$\Rightarrow P(t) = e^{-\int_0^t \lambda(t).dt} \quad (1.12)$$

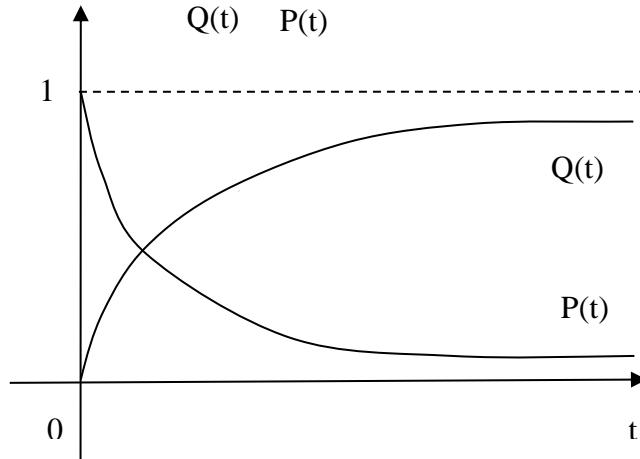
Công thức (1.12) cho phép tính được độ tin cậy của phần tử trong trường hợp không phục hồi khi đã biết cường độ hỏng hóc  $\lambda(t)$ , mà giá trị của  $\lambda(t)$  này xác định được nhờ phương pháp thống kê quá trình hỏng hóc của phần tử trong quá khứ [89].

Trong các hệ thống hiện nay thường sử dụng điều kiện  $\lambda(t) = \lambda =$  hằng số ( $\lambda$  tương đối nhỏ), điều này thực hiện được nhờ bảo quản định kỳ. Khi đó cường độ hỏng hóc là giá trị trung bình số lần sự cố xảy ra trong một đơn vị thời gian.

Khi đó, Công thức (1.12) trở thành:  $P(t) = e^{-\lambda t}$  (1.13)

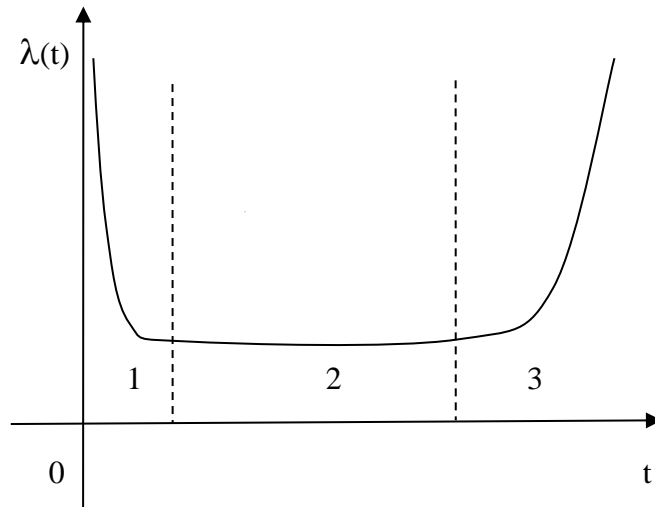
$$\text{Và: } Q(t) = 1 - e^{-\lambda t}, q(t) = \lambda e^{-\lambda t}$$

Biểu diễn mối quan hệ giữa các thông số trên như Hình 1.4 sau:



Hình 1.4: Tương quan giữa độ tin cậy và độ hỏng của phần tử

Qua Hình 1.4 có thể thấy, tại thời điểm đầu tiên ( $t=0$ ), độ tin cậy của phần tử luôn là 1, tức là hệ ở trạng thái sẵn sàng làm việc theo thông số của nhà sản xuất. Theo thời gian hoạt động, độ tin cậy này sẽ giảm theo hàm thời gian  $t$  đến khi hỏng hẳn. Ngược lại, xác suất sự cố xảy ra của phần tử là 0 và sẽ tăng dần theo thời gian, hai hàm  $P(t)$  và  $Q(t)$  luôn tỉ lệ nghịch với nhau.



Hình 1.5: Biểu diễn cường độ hỏng của phần tử

Theo biểu đồ của Hình 1.5 ta thấy được mối quan hệ của cường độ hỏng hóc  $\lambda(t)$  theo thời gian. Đường cong của cường độ hỏng hóc  $\lambda(t)$  được chia làm ba miền:

*Miền thứ nhất:* Mô tả thời kỳ “chạy thử”, những hỏng hóc ở giai đoạn này thường do lắp ráp, vận chuyển. Tuy giá trị ở giai đoạn này cao nhưng thời gian kéo dài ít, giảm dần và nhờ chế tạo, nghiệm thu có chất lượng nên giá trị cường độ hỏng hóc  $\lambda(t)$  ở giai đoạn này có thể giảm nhiều.

*Miền thứ 2:* Mô tả giai đoạn sử dụng hữu ích, cũng là giai đoạn chủ yếu của tuổi thọ các phần tử. Ở giai đoạn này, các sự cố thường xảy ra ngẫu nhiên, đột ngột do nhiều nguyên nhân khác nhau, vì vậy thường giả thiết cường độ hỏng hóc  $\lambda(t)$  bằng hằng số.

*Miền thứ ba:* Mô tả giai đoạn cuối của vòng đời phần tử theo thời gian, cường độ hỏng hóc  $\lambda(t)$  tăng dần, đó là điều tất yếu xảy ra sự cố khi  $t \rightarrow \infty$ .

### 1.1.2. Một số thuật ngữ liên quan độ tin cậy

Trong lý thuyết về độ tin cậy, một số thuật ngữ thường được sử dụng như: MTTF, MTTR, MTBF... đây là các thuật ngữ về độ tin cậy dựa trên các phương pháp và quy trình dự đoán vòng đời của các phần tử hay hệ thống.

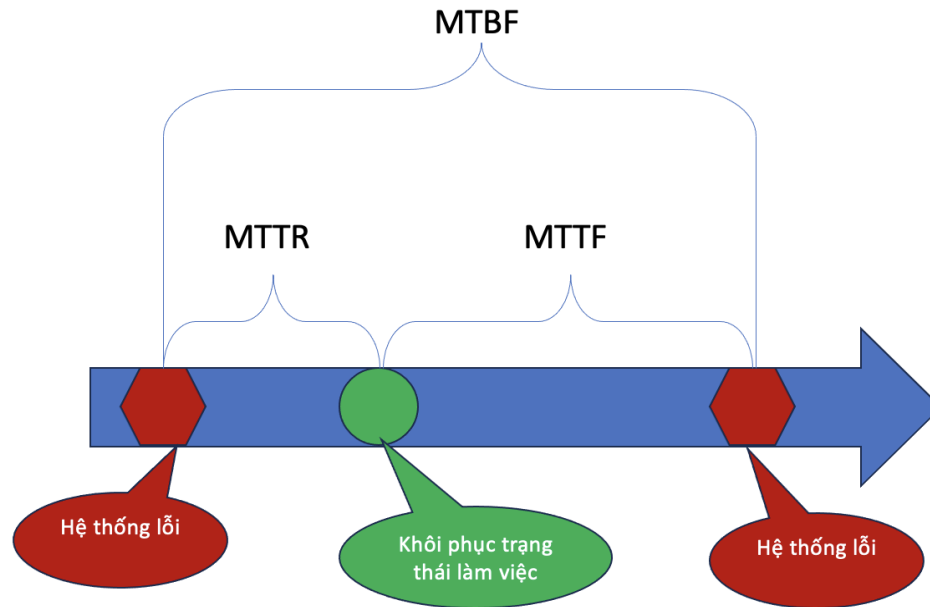
**Bảng 1.1: Một số thông số về độ tin cậy thường được sử dụng**

Thuật ngữ	Ý nghĩa	Diễn giải
<b>Mean Time Between Failure (MTBF)</b>	Thời gian trung bình giữa các lần hỏng hóc	Là số lượng lỗi trên một triệu giờ của một thiết bị, chỉ số này thường được sử dụng để ước tính thời gian trung bình giữa các lỗi xảy ra trong hệ thống. MTBF cho biết tuổi thọ của thiết bị hoặc khả năng hoạt động và điều này rất quan trọng trong quá trình ra quyết định của người dùng cuối. MTBF có ý

		<p>nghĩa đối với các ngành công nghiệp và nhà tích hợp hơn là đối với người tiêu dùng. Hầu hết người tiêu dùng đều bị định hướng về giá và sẽ không xem xét MTBF cũng như không có sẵn dữ liệu thường xuyên. Mặt khác, khi phải cài đặt các thiết bị như bộ chuyển đổi phương tiện hoặc bộ chuyển mạch vào các ứng dụng quan trọng, MTBF trở nên rất quan trọng. Ngoài ra, MTBF có thể là một thông tin cần thiết thường được sử dụng trên các thiết bị có khả năng hồi phục sau khi gặp sự cố.</p>
<p><b>Mean Time To Failure (MTTF)</b></p>	<p>Thời gian hỏng trung bình</p>	<p>MTTF là thước đo cơ bản về độ tin cậy của các hệ thống không thể phục hồi, điều này ngược lại với giá trị MTBF. Đây được coi là thời gian trung bình dự kiến cho đến khi xảy ra hư hỏng đầu tiên của một thiết bị. MTTF là một giá trị thống kê và được coi là giá trị trung bình trong một khoảng thời gian dài và một số lượng lớn các đơn vị.</p>
<p><b>Mean Time To Repair (MTTR)</b></p>	<p>Thời gian sửa chữa trung bình</p>	<p>MTTR được sử dụng để chỉ thời gian cần thiết để sửa chữa một module phần cứng bị lỗi. Trong một hệ thống vận hành, sửa chữa thường có nghĩa là thay thế một bộ phận phần cứng bị lỗi. Việc mất quá nhiều thời gian để sửa chữa một thiết bị về lâu dài sẽ làm tăng chi phí lắp đặt do thời gian ngừng hoạt động cho đến khi bộ phận mới được giao đến và khoảng thời gian có thể cần thiết để lên lịch lắp đặt. Để hạn chế ảnh hưởng đối với các thiết bị có MTTR cao, khi thiết kế hệ thống, ta thường đưa vào các</p>

		thiết bị dự phòng để có thể lắp đặt sản phẩm thay thế nhanh chóng.
--	--	--

Mối liên hệ giữa các thông số độ tin cậy trong Bảng 1.1 được biểu diễn trong Hình 1.6:



Hình 1.6: Mối liên hệ giữa các thống số MTBF, MTTR và MTTF.

Mối liên quan giữa giá trị MTTF và cường độ hỏng của phần tử được xác định theo công thức [64, 71]:

$$MTTF = \int_0^{+\infty} P(t) dt \quad (1.14)$$

Theo Công thức (1.13), ta có:

$$MTTF = \int_0^{+\infty} P(t) dt = \int_0^{+\infty} e^{-\lambda.t} dt = \frac{1}{\lambda} \quad (1.15)$$

Trong khi đó, giá trị MTBF được xác định theo:

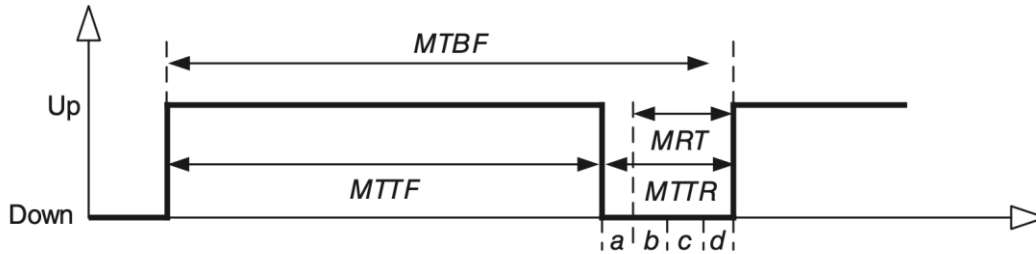
$$MTBF = MTTF + MTTR$$

Và giá trị MTTR bao gồm các thông số như:

- a) Thời gian phát hiện sự kiện hỏng;

- b) Thời gian trải qua trước khi bắt đầu sửa;
- c) Thời gian để sửa;
- d) Thời gian trước khi phân tử được đưa trở lại vận hành.

Trong đó, các thông số a) đến d) được thể hiện trong Hình 1.7 như sau:



Hình 1.7: Mối liên hệ giữa MTTF và MTBF [71]

Đối với phân tử không phục hồi thì  $MTBF = MTTF$ , đó là lý do chỉ tồn tại thông số MTBF ở các phân tử không phục hồi. Trong các hệ thống hay thiết bị kỹ thuật, các phân tử thường tồn tại ở hai dạng là: phân tử có phục hồi và phân tử không phục hồi [64, 89].

Phục hồi là quá trình phát hiện và khắc phục hư hỏng để thiết lập lại khả năng làm việc của phân tử hoặc hệ thống. Việc thiết lập lại khả năng làm việc của phân tử, trong trường hợp hư hỏng đang xét, có thể là hoàn toàn vô ích hoặc không thể tiến hành được; khi đó phân tử được coi là không phục hồi. Ví dụ, các đèn điện tử, các linh kiện bán dẫn.

Ngược lại, phân tử mà khả năng làm việc của nó có thể được khôi phục lại trong trường hợp hư hỏng được gọi là sản phẩm có phục hồi. Tuy nhiên có trường hợp tùy theo từng giai đoạn hay tùy theo mục đích sử dụng mà một sản phẩm có thể được coi là có phục hồi hay không phục hồi. Như một máy chủ dịch vụ có thể coi là hệ thống không phục hồi, khi gặp sự cố máy chủ đó sẽ ngưng làm việc và có thể dẫn đến ảnh hưởng cả dây truyền liên quan. Trường hợp máy chủ được thiết

lập và cấu hình để có thể tự khởi động lại khi gặp sự cố và sau khi khởi động, các dịch vụ hoạt động trên đó được kích hoạt tự động và trở về trạng thái hoạt động như ban đầu, khi đó hệ thống là có khả năng phục hồi.

### 1.1.3 Bài toán đánh giá độ tin cậy của hệ thống mạng

Bài toán đánh giá độ tin cậy của mạng đã được nghiên cứu và áp dụng thành công cho nhiều hệ thống trong thế giới thực, như hệ thống thông tin [13, 29, 44, 81], hệ thống sản xuất [33, 45, 86], hệ thống viễn thông [41, 47], hệ thống truyền tải điện và hệ thống giao thông [43, 55, 81, 86]. Các hệ thống mạng được sử dụng để mô phỏng các cấu trúc, trong đó các nút mạng thực hiện truyền thông dữ liệu với nhau, thường được mô hình hóa và biểu diễn bằng biểu đồ hoặc đồ thị (vô hướng hoặc có hướng). Các đỉnh đồ thị biểu diễn các nút (thiết bị) mạng, trong khi các cạnh biểu thị sự kết nối giữa các nút đó.

Độ tin cậy của các phần tử trong mạng có thể được phân chia thành ba bài toán chính: a) Bài toán độ tin cậy của hai thiết bị đầu cuối (tức là hai nút được coi là thiết bị đầu cuối, thường gọi là *two-terminal*); b) Bài toán độ tin cậy của  $k$  thiết bị (tức là  $k$  nút được coi là thiết bị đầu cuối, gọi là *k-terminal*) và c) Bài toán độ tin cậy của tất cả thiết bị trong mạng (tức là tất cả các nút đều được coi là thiết bị đầu cuối, gọi là *all-terminal*).

Tuy nhiên, bài toán độ tin cậy của hai thiết bị đầu cuối có ý nghĩa rất quan trọng vì đây là nền tảng của các bài toán còn lại. Độ tin cậy của cặp thiết bị đầu cuối là xác suất mà hai nút được chọn (nút đầu cuối) có thể giao tiếp thông qua ít nhất một đường dẫn của các cạnh làm việc.

Độ tin cậy của hai nút mạng bất kỳ được đo bằng xác suất tồn tại ít nhất một đường đi giữa hai nút đó. Trong mô hình mạng này, giả định rằng các nút và cạnh chỉ có hai trạng thái (Binary-State Network): làm việc hoặc không làm việc. Mặc

dù các nút được giả định là hoàn toàn đáng tin cậy và trạng thái của các cạnh là độc lập, tuy nhiên, việc đánh giá độ tin cậy của mạng ngẫu nhiên được chứng minh là một bài toán NP-khó [6, 14]. Các phương pháp đánh giá độ tin cậy sử dụng nhiều công cụ khác nhau để thiết lập mô hình cho hệ thống mạng và tính toán chỉ số độ tin cậy.

Một số thuật toán đã được đề xuất để tính độ tin cậy giữa hai thiết bị đầu cuối, theo đó có thể được phân loại thành ba cách tiếp cận chính:

- Cách tiếp cận một: Sử dụng kỹ thuật Sum-of-Disjoint Products (SDP), trong cách tiếp cận này, các mối liên kết trong đồ thị được liệt kê và sau đó sẽ áp dụng kỹ thuật SDP để tách rời các điều kiện ràng buộc trước khi thực hiện tính độ tin cậy [30, 82, 84]. Theo đánh giá, kỹ thuật này dễ dàng thực hiện và cài đặt; tuy nhiên, hiệu suất tính toán là vấn đề cần quan tâm, vì thời gian thực hiện tăng lên khi áp dụng trên các mạng lớn do sự tăng trưởng theo cấp số nhân của các cạnh trong đồ thị của mạng.
- Cách tiếp cận hai (sử dụng phương pháp xấp xỉ): Các phương pháp gần đúng được đề xuất nhằm giảm thời gian thực hiện và độ phức tạp tính toán. Các phương pháp này sử dụng giới hạn dưới và giới hạn trên của độ tin cậy hoặc sử dụng kỹ thuật lấy mẫu theo phương pháp Monte Carlo [8, 21, 57] để ước tính kết quả, thuật toán giới hạn lần đầu tiên được đề xuất bởi [39, 63].
- Cách tiếp cận ba (Kỹ thuật Binary Decision Diagram (BDD)): Để khắc phục điểm yếu của cách tiếp cận thứ nhất, các phương pháp dựa trên BDD đã được đề xuất trong nghiên cứu [4]. BDD được coi là cấu trúc dữ liệu hiệu quả để lưu trữ số lượng lớn các thuật ngữ Boolean. Năm 1999, Kuo và cộng sự đã phát triển một thuật toán dựa trên các BDD có thứ tự để tránh các tính toán dư thừa bằng cách phát hiện các sơ đồ con đẳng cấu [70].

Như vậy, với các hệ thống mạng có cấu trúc phức tạp với số lượng nút lớn, việc đánh giá độ tin cậy của hệ thống sẽ trở lên khó khăn hơn. Phương pháp sử dụng kỹ thuật SDP có ưu điểm trong việc cài đặt và triển khai thuật toán, hạn chế về tốc độ và hiệu suất của SDP có thể cải tiến bằng các phương pháp như song song hóa quá trình thực hiện tính toán với sự trợ giúp của thiết bị phần cứng mới như GPU.

## **1.2. Tổng quan về các phương pháp tính độ tin cậy của hệ thống**

Trong các lĩnh vực kỹ thuật và công nghiệp, việc đảm bảo độ tin cậy của hệ thống là vô cùng quan trọng để duy trì hiệu suất hoạt động ổn định và giảm thiểu rủi ro. Các phương pháp tính độ tin cậy hệ thống giúp xác định và phân tích các yếu tố có thể gây ra hỏng hóc hoặc sự cố, từ đó đưa ra các biện pháp phòng ngừa và cải thiện. Hiện nay có nhiều phương pháp được các nhà khoa học nghiên cứu và sử dụng trong việc phân tích độ tin cậy của các hệ thống trong nhiều lĩnh vực khác nhau như: vận hành lưới điện và truyền tải điện [8, 32, 48, 49, 57], hệ thống mạng máy tính [42, 70, 83-85, 87]. Dựa trên kết quả tính toán và giả thiết để mô hình hóa các tình huống sự cố có thể xảy ra trong thực tiễn, từ đó đánh giá xác suất xuất hiện lỗi nhằm đưa ra các chiến lược để tối ưu hóa các chiến lược bảo trì và sửa chữa. Điều này không chỉ giúp tăng tuổi thọ và hiệu suất của hệ thống mà còn giảm thiểu chi phí bảo trì và ngừng hoạt động không mong muốn, đảm bảo an toàn cho người sử dụng và môi trường xung quanh. Sự cần thiết của các phương pháp này càng trở nên cấp bách trong bối cảnh các hệ thống ngày càng phức tạp và yêu cầu cao về độ tin cậy.

### **1.2.1. Phương pháp liệt kê trạng thái (State Enumeration - SE)**

Trong các phương pháp tiếp cận độ tin cậy của một hệ thống mạng, có hai phương pháp được sử dụng rất phổ biến đó là: dựa trên tập hợp không cắt (non-

cut set based method) và dựa trên tập hợp cắt (cut set based method). Bảng liệt kê trạng thái (SE) là một kỹ thuật dựa trên phương pháp sử dụng tập hợp không cắt, đây là thuật toán được coi là đơn giản nhất hiện có để đánh giá độ tin cậy của các hệ thống mạng. Ý tưởng chính trong kỹ thuật SE đó là liệt kê tất cả các trạng thái có thể tồn tại của hệ thống [32]. Các trạng thái của hệ thống được liệt kê đến mức đạt được ngưỡng xác suất nhất định hoặc mức lỗi xác định [35].

Phương pháp SE được sử dụng tốt nhất trong việc đánh giá độ tin cậy của các hệ thống lưới điện và truyền tải điện [49]. Trong nghiên cứu của Billinton [8] đã kết hợp phương pháp SE với mô phỏng Monte Carlo để giải quyết bài toán về độ tin cậy của hệ thống truyền tải. Nhóm của Rei [57] cũng sử dụng SE và mô phỏng Monte Carlo cho việc xác định độ tin cậy của hệ thống lưới điện Brazil, nghiên cứu cho thấy sự hoạt động hiệu quả với trường hợp dự phòng đơn lẻ phương pháp này trở nên rất khó giải quyết đối với trường hợp sử dụng nhiều dự phòng trong hệ thống. Chính vì vậy, phiên bản cải tiến của phương pháp SE cho phép tính toán xác suất trong trường hợp hệ thống có nhiều trạng thái đã được nghiên cứu và áp dụng trong đánh giá độ tin cậy của các hệ thống truyền tải điện năng [35, 48].

Có thể thấy rằng, phạm vi áp dụng của phương pháp SE được giới hạn ở phân tích hệ thống điện trong các trường hợp dự phòng ở một cấp độ đơn lẻ. Biểu thức độ tin cậy thu được khi áp dụng phương pháp SE khó sử dụng, ngay cả đối với các mạng nhỏ. Ngoài ra, đối với các biểu thức độ tin cậy dựa trên xác suất lỗi của từng thành phần thường mất nhiều thời gian tính toán. Độ phức tạp tính toán của phương pháp SE là một vấn đề nghiêm trọng đối với các nhà nghiên cứu khi trạng thái hệ thống cũng như số lượng thành phần trong hệ thống tăng lên. Điều

đó dẫn đến, phương pháp SE này không được sử dụng phổ biến khi đánh giá độ tin cậy của các hệ thống.

### 1.2.2. Phương pháp cắt cực tiểu

Phương pháp cắt cực tiểu (Minimal Cut – MC) là một kỹ thuật trong lý thuyết độ tin cậy và phân tích hệ thống được sử dụng để xác định các tập hợp phần tử mà nếu tất cả các phần tử trong tập hợp này bị hỏng, thì toàn bộ hệ thống sẽ ngừng hoạt động. Đây là một công cụ hữu ích để đánh giá và cải thiện độ tin cậy của hệ thống bằng cách xác định các điểm yếu chính trong cấu trúc của nó. Nếu chúng ta loại bỏ một số cạnh khỏi mạng và nếu điều đó làm ngắt kết nối từ đỉnh nguồn  $s$  đến đỉnh đích  $t$  thì tập hợp các cạnh đó được gọi là một đường cắt [42]. Trong nghiên cứu của Younes và Girgis [85] đã sử dụng phương pháp MC để thực hiện tính độ tin cậy trong hệ thống mạng các máy tính dựa trên xác suất thành công của các đường dẫn. Thuật toán bao gồm hai bước: đầu tiên tìm tập hợp đường dẫn tối thiểu trong mạng; sau đó tính toán độ tin cậy dựa trên xác suất liên kết của các liên kết trong đường dẫn tối thiểu ở bước một để tính độ tin cậy mạng của mạng máy tính. Trong khi đó, nghiên cứu của Yeh [83] đề xuất một phương pháp tính toán mới nhằm tìm ra các mức cắt cực tiểu (MC) trong hệ thống mạng từ các nút và cạnh ban đầu bằng cách bổ sung các nút mới để tạo điều kiện mở rộng hệ thống hiện có. Đây cũng là một phương pháp hữu dụng khi cần xác định các mức cắt cực tiểu trong hệ thống mạng phức tạp, từ đó giúp xác định độ tin cậy của hệ thống.

Mặc dù vậy, vẫn còn một số hạn chế với thuật toán tập hợp cắt tối thiểu. Phương pháp MC chỉ hoạt động hiệu quả đối với bài toán tìm độ tin cậy của hai thiết bị đầu cuối và số lượng các nút mạng là không lớn. Với bài toán xác định độ tin cậy của nhiều thiết bị đầu cuối trong mạng, khi kích thước tập hợp cắt tăng theo

cấp số nhân thì phương pháp MC không còn hoạt động hiệu quả. Ngoài ra, kích thước tập cắt trở nên khó điều chỉnh đối với các mạng lớn để thực hiện các phép tính về độ tin cậy [24]. Cuối cùng, bài toán tính toán độ tin cậy dựa trên thước đo xác suất của tập cắt là một bài toán NP khó, do đó quá trình giải đòi hỏi phương pháp tính toán chuyên sâu.

### 1.2.3. Phương pháp tổng sản phẩm rời rạc

Phương pháp tổng sản phẩm rời rạc (Sum of Disjoint Product – SDP) là một dạng dẫn xuất của phương pháp sử dụng tập cắt. Trong phương pháp SDP, bài toán tính độ tin cậy của hai phần tử trong hệ thống mạng được định hình dưới dạng bài toán NP khó, với xác suất tồn tại ít nhất một đường kết nối từ nút nguồn đến nút đích [84].

Phương pháp SDP được thực hiện theo hai giai đoạn: Giai đoạn một bao gồm việc tìm kiếm tất cả các đường dẫn tổng hợp, bao gồm đường dẫn tối thiểu tồn tại từ nút nguồn  $s$  đến nút đích  $t$ ; Giai đoạn hai thực hiện việc tách các đường dẫn tổng hợp đã tìm được để đánh giá độ tin cậy của hệ thống [82]. Trong nghiên cứu của Lin và cộng sự [46] đã đánh giá tập hợp đường dẫn tối thiểu và sau đó áp dụng một cách đệ quy phương pháp SDP để tính toán độ tin cậy của mạng máy tính, trong đó có xét đến tỷ lệ lỗi liên quan đến quá trình truyền. Phương pháp SDP được ứng dụng chủ yếu trong việc đánh giá độ tin cậy của luồng ngẫu nhiên trong các hệ thống mạng [82]. Tuy nhiên, việc xây dựng đường dẫn tổng hợp và quá trình tách rời đều là những bài toán được xếp vào dạng bài toán NP khó. Mức độ phức tạp liên quan đến việc giải các bài toán NP khó này dẫn đến sự kém hiệu quả của phương pháp SDP này trong việc đánh giá độ tin cậy hệ thống. Thời gian cần thiết để tính toán các số hạng rời rạc tăng theo cấp số nhân khi số lượng đường đi hoặc tập hợp cắt tối thiểu tăng lên.

#### 1.2.4. Phương pháp biểu đồ quyết định nhị phân

Binary Decision Diagram (BDD) là một cấu trúc dữ liệu được sử dụng để biểu diễn các hàm Boolean. Một BDD là một đồ thị không chu trình có hướng (DAG - Directed Acyclic Graph) trong đó các đỉnh biểu diễn các biến Boolean, và các cạnh thể hiện các giá trị của các biến đó (0 hoặc 1). Mỗi đường dẫn từ gốc đến lá của BDD biểu diễn một tổ hợp các biến và giá trị của chúng, và giá trị của hàm Boolean tại đầu ra. BDD được biểu diễn dưới dạng hàm như sau:

$$f = x \cdot f_{x=1} + \bar{x} \cdot f_{x=0}$$

Trong đó  $f$  là hàm Boolean cho biến ngẫu nhiên  $x$  trong một hệ thống, các giá trị “0” và “1” của  $x$  thể hiện cho trạng thái làm việc và hỏng của hệ thống mạng [87]. BDD được coi là một cấu trúc dữ liệu hiệu quả để lưu trữ số lượng lớn các thuật ngữ Boolean, vì vậy được ứng dụng trong các nghiên cứu về độ tin cậy của hệ thống mạng [28, 36, 50, 70, 79, 87].

Nghiên cứu của Xing [29] đã đề xuất một cách tiếp cận mới dựa trên BDD để đánh giá độ tin cậy của mạng bằng cách kết hợp các vấn đề về độ phủ không đầy đủ (Incomplete Coverage - IPC) với các nguyên nhân gây ra lỗi phổ biến (Common Cause Failures - CCF). Hardy [28] đã kết hợp kỹ thuật phân rã mạng với BDD để tính toán độ tin cậy của  $k$  thiết bị đầu cuối trong hệ thống mạng. Kawahara và nhóm [36] đã tính toán độ tin cậy của mạng bằng cách sử dụng phương pháp BDD trên cơ sở xem xét lỗi xảy ra ở cả cạnh và nút mạng. BDD có hiệu quả trong việc cung cấp giải pháp cho các mạng có tính chất nhị phân của các nút, điều này làm cho quá trình tính toán trở nên nhanh hơn đối với các mạng có số lượng nút cao. Tuy nhiên, không thể phân tích một phần độ tin cậy của hệ thống bằng cách sử dụng BDD và quá trình thực hiện khá phức tạp đối với BDD trong trường hợp các luồng ngẫu nhiên và các mạng phụ thuộc lẫn nhau.

### **1.2.5. Phương pháp Simple Algorithm For Computing Network Reliability (SACNR)**

SACNR là một thuật toán tính toán độ tin cậy của hệ thống mạng theo xác suất thành công của các liên kết trên các đường dẫn tối thiểu trong mạng, được nhóm tác giả Younes giới thiệu trong [85]. Nội dung nghiên cứu đã đưa ra một thuật toán để tìm kiếm tất cả các đường dẫn tối thiểu tồn tại trong mạng trước khi sử dụng để xác định độ tin cậy của mạng. Thuật toán dựa trên mối quan hệ sử dụng xác suất kết hợp các đường dẫn tối thiểu của mạng để tính được độ tin cậy của mạng. SACNR bao gồm hai giai đoạn chính: Giai đoạn xác định đường dẫn tối thiểu và giai đoạn tính toán độ tin cậy.

Giai đoạn đầu tiên sử dụng các liên kết của mạng và xác suất của chúng, sau đó thực hiện thuật toán để xác định các đường dẫn tối thiểu trong mạng. Giai đoạn thứ hai thực hiện thuật toán được đưa ra để tính toán độ tin cậy của mạng. Kết quả của nghiên cứu được áp dụng như là công cụ tính toán độ tin cậy của một mô hình mạng bất kỳ với các đỉnh, các cạnh có xác suất hoạt động cụ thể.

### **1.2.6. Một số nhận xét**

Bài toán tính độ tin cậy của hệ thống mạng là một trong các bài toán có tính ứng dụng cao trong thực tiễn. Trong đó bài toán độ tin cậy giữa hai điểm nguồn và đích của mạng thu hút nhiều sự quan tâm của giới nghiên cứu, do đó được coi như bài toán gốc của các nghiên cứu khác trong mạng. Để tính toán được độ tin cậy giữa hai điểm đầu cuối, sự cần thiết phải tìm ra đường đi giữa hai điểm đó và các phương pháp như MC, SE, SACRN thực sự hiệu quả để tìm ra đường đi tối thiểu, từ đó tính toán độ tin cậy cho đường đi.

Tuy nhiên, với sự phát triển và mở rộng của các hệ thống mạng như hiện nay, số lượng nút mạng và cạnh nối các điểm tăng lên, thì các thuật toán cổ điển

đề cập ở trên như SDP, BDD sẽ gặp trở ngại đáng kể trong việc tính toán và do đó hiệu suất hoạt động bị ảnh hưởng. Ngoài ra, các phương pháp tính toán được biết đến đều sử dụng cơ chế tuần tự, việc xác định độ tin cậy của các tuyến đường tối thiểu sau khi xây dựng hoàn toàn có thể được xử lý độc lập và song song hóa nhằm giảm thời gian tính toán. Hiện nay, với sự phát triển của phần cứng máy tính, các thuật toán song song được áp dụng rất phổ biến [54, 74] và có sự hỗ trợ của những thiết bị đồ họa mạnh như NVIDIA hay bộ vi xử lý đa nhân của Intel, AMD [11]. Vì vậy, việc ứng dụng tính toán song song vào bài toán tính độ tin cậy giữa các nút mạng trở nên thực tế và tính ứng dụng cao.

### **1.3. Tổng quan về các phương pháp đánh giá độ tin cậy của hệ thống**

Đánh giá độ tin cậy là một khía cạnh quan trọng trong kỹ thuật, nhằm đảm bảo rằng các hệ thống và bộ phận thực hiện đúng chức năng dự kiến trong các điều kiện nhất định trong một khoảng thời gian cụ thể. Mục tiêu chính của công việc này là dự đoán và giảm thiểu các lỗi tiềm ẩn, cải thiện hiệu suất tổng thể và đảm bảo an toàn cho hệ thống trong quá trình hoạt động trước khi đưa vào triển khai thực tiễn. Hiện có nhiều phương pháp khác nhau được sử dụng để đánh giá độ tin cậy, mỗi phương pháp có trọng tâm và lĩnh vực ứng dụng riêng. Một số phương pháp chi tiết được sử dụng trong đánh giá độ tin cậy:

#### **1.3.1. Phương pháp mô phỏng Monte Carlo**

Phương pháp mô phỏng Monte Carlo (hay Monte Carlo simulation – MCS) là một kỹ thuật lấy mẫu được sử dụng phổ biến để ước lượng độ tin cậy của mạng do sự đơn giản trong việc tính toán và linh hoạt trong quá trình sử dụng. MCS thường được sử dụng để giải quyết các vấn đề phức tạp mà không thể giải bằng các phương pháp phân tích truyền thống. Trong một số nghiên cứu gần đây, các nhà khoa học đã kết hợp MCS với thuật toán tính độ tin cậy như: trong nghiên cứu

của Khadiri [21], các tác giả đã sử dụng MCS để tính toán độ tin cậy đối của luồng lưu lượng tối đa với thời gian thực hiện và sự biến đổi ngẫu nhiên của các đường liên kết trong mạng. Nghiên cứu của Moreno [59] đã thực hiện kết hợp giữa MCS với phương pháp Cellular Automata (CA) (là một hệ thống rời rạc trạng thái hữu hạn để mô phỏng các hệ thống động, cũng được sử dụng trong tính toán độ tin cậy của hệ thống) để tìm kiếm đường đi giữa hai điểm đầu cuối trong mạng với độ tin cậy cực đại, dựa trên kết quả nghiên cứu, tác giả đã thực hiện việc đánh giá độ tin cậy của mạng máy tính. Trong nghiên cứu của Forghani-elahabad và Kagan [24], các tác giả đã kết hợp MCS với phương pháp tập cắt tối thiểu (Minimal cut-set method) để tính toán độ tin cậy của mạng với luồng lưu lượng ngẫu nhiên, đồng thời sử dụng MCS trong việc định lượng tính chất ngẫu nhiên của các thành phần khác nhau trong một hệ thống.

Nghiên cứu của tác giả Murray và cộng sự [52] đã sử dụng công cụ ước tính Monte Carlo để giảm phương sai cho kết quả thuật toán phân tách (mô phỏng tập hợp con) của hiệu suất mạng. Ngoài ra, MCS có thể thu thập được tính năng động của hệ thống, Ramirez-Marquez và Coit [10] đã áp dụng thành công MCS đối với bài toán về độ tin cậy của hai thiết bị đầu cuối trong hệ thống với các nút đa trạng thái.

Mặc dù được ứng dụng rộng rãi trong việc giải nhiều dạng bài toán khác nhau, tuy nhiên phương pháp MCS có một số hạn chế trong việc đánh giá độ tin cậy của hệ thống như: Đối với các hệ thống lớn và phức tạp, thời gian cần thiết mô phỏng của MCS có thể rất dài, làm cho việc sử dụng phương pháp này trở nên không khả thi. Trong quá trình thực hiện, MCS yêu cầu một không gian giải pháp đủ lớn, tuy nhiên điều này có thể gây ra khó khăn trong việc quản lý và tính toán; MCS gặp khó khăn trong việc đánh giá chính xác các xác suất lỗi nhỏ, dẫn đến kết

quả không đủ tin cậy cho các hệ thống yêu cầu độ chính xác cao; Sự hội tụ của giải pháp MCS cho các mạng đa trạng thái thường chậm, điều này hạn chế khả năng ứng dụng của MCS trong việc đánh giá độ tin cậy của các hệ thống phức tạp; Cuối cùng, MCS thường yêu cầu một lượng lớn tài nguyên tính toán, đặc biệt là đối với các hệ thống phức tạp với nhiều trạng thái và biến số.

Những hạn chế này khiến cho phương pháp Monte Carlo Simulation không phải lúc nào cũng là lựa chọn tối ưu cho việc đánh giá độ tin cậy của các hệ thống phức tạp.

### **1.3.2. Phương pháp sử dụng chuỗi Markov (Markov chain)**

Mô hình Markov là một công cụ mạnh mẽ để mô phỏng nhiều quy trình ngẫu nhiên đa trạng thái. Phương pháp này sử dụng cách tiếp cận không gian trạng thái để mô hình hóa chuỗi các quá trình ngẫu nhiên hoặc các sự kiện ngẫu nhiên diễn ra trong hệ thống điện. Để áp dụng mô hình Markov vào phân tích độ tin cậy của hệ thống điện, cần phải mô hình hóa rõ ràng tất cả các trạng thái có thể có và sự chuyển đổi giữa chúng. Khi mô hình hóa hệ thống điện, ma trận chuyển trạng thái có thể trở nên rất lớn. Gánh nặng tính toán khi phải đảo ngược ma trận chuyển trạng thái thường hạn chế khả năng áp dụng mô hình Markov cho các hệ thống tương đối nhỏ như một trạm biến áp.

Chuỗi Markov là một phương pháp tương đối mới để ước tính độ tin cậy của hệ thống mạng thông qua việc đánh giá xác suất của các trạng thái trong hệ thống và do đó, phù hợp với các hệ thống đa trạng thái. Koorosh và các cộng sự [3] đã sử dụng mô hình chuỗi Markov trong việc đánh độ tin cậy và ước lượng thời gian xảy ra hỏng của các động cơ điều khiển cánh quạt trên thiết bị bay không người lái (UAV). Với việc giả định các trạng thái có thể xảy ra trên hệ thống, nghiên cứu đã đề xuất mô hình có khả năng phát hiện lỗi, từ đó có thể khôi phục

lại trạng thái hoạt động của hệ thống nhằm giảm thiểu tác động của lỗi đến sự vận hành của thiết bị.

Nghiên cứu của nhóm tác giả Kim [37] sử dụng mô hình chuỗi Markov để thiết kế và phân tích độ tin cậy áp dụng trên hệ thống với các phần tử không đồng nhất và không có khả năng phục hồi, với phân bố thời gian xảy ra lỗi theo kiểu pha. Nghiên cứu đã đề xuất mô hình nhằm giải quyết vấn đề phân bổ dự phòng, qua đó cải thiện được mức độ dư thừa các phần tử trong hệ thống. Xem xét chiến lược dự phòng để tối đa hóa độ tin cậy của hệ thống.

Trong khi đó, các nghiên cứu [26, 27, 78, 80] tập trung vào việc phân tích và đánh giá độ tin cậy trên hệ thống với các phần tử có phục hồi nhằm tối ưu hóa chiến lược dự phòng cho hệ thống, tăng cường tính sẵn sàng và nâng cao độ tin cậy cho hệ thống trong quá trình sử dụng.

Có thể thấy rằng, phương pháp chuỗi Markov có nhiều tính ứng dụng trong các nghiên cứu thực tiễn dựa trên việc mô phỏng các trạng thái tồn tại khác nhau của hệ thống, hay các phần tử của hệ thống, với xác suất hỏng tuân theo mô hình ngẫu nhiên. Tuy nhiên, phương pháp này có hai vấn đề chính đó là: việc xây dựng chuỗi Markov cho các trạng thái hệ thống là một nhiệm vụ tương đối phức tạp, đặc biệt trong các hệ thống đa trạng thái cùng hoạt động; Khả năng bùng nổ không gian trạng thái trong đó các trạng thái của hệ thống tăng theo cấp số nhân khi số lượng thành phần tăng lên khiến cho việc tính toán trở nên kém hiệu quả.

### **1.3.3. Phương pháp sử dụng mạng Bayesian**

Mạng Bayesian (Bayesian Network - BN) là một mô hình đồ thị được sử dụng để biểu diễn và suy luận về các mối quan hệ xác suất giữa các biến ngẫu nhiên. Đây là một công cụ mạnh mẽ trong việc xử lý các vấn đề liên quan đến sự không chắc chắn và suy luận dựa trên bằng chứng. Mạng BN được biểu diễn dưới

dạng một đồ thị có hướng không chu trình dựa trên mô hình xác suất và tham số của cấu trúc [18, 51]. Trong BN, các tham số được mô hình hóa dưới dạng các biến ngẫu nhiên dựa trên phân bố xác suất [38]. Boudali và Dugan [9] đã xác định một BN rời rạc để phân tích độ tin cậy của các hệ thống động phức tạp và kết quả cho thấy sự hiệu quả hơn so với các phương pháp hiện có trong việc tránh sự trùng lặp của các nút khi tiến hành phân tích nguyên nhân lỗi xảy ra. Một khía cạnh khác của BN để phân tích độ tin cậy là ứng dụng của nó trong việc đánh giá khả năng phục hồi dựa trên thành phần của hệ thống. Daemi và Ibrahim [18] đã so sánh hai phương pháp BN và MCS để đánh giá phân tích độ tin cậy ở cấp độ thành phần và chứng minh rằng MCS có thể cho kết quả tốt hơn với hệ thống có số lượng mẫu thử nhỏ hơn. Mạng BN cũng được áp dụng trong các bài toán với thành phần có khả năng phục hồi [7, 31] nhằm ước tính hiệu suất phục hồi ở cấp độ thành phần trong hệ thống. Tuy nhiên, việc lựa chọn hàm phân phối xác suất tiên nghiệm là vấn đề mang tính chủ quan trong việc áp dụng phương pháp niềm tin Bayes.

#### **1.3.4. Phương pháp sử dụng phân tích cây sai**

Phương pháp sử dụng Phân tích cây lỗi (Fault Tree Analysis - FTA) đã được sử dụng rộng rãi để đánh giá độ tin cậy của các hệ thống kỹ thuật từ lâu. Ưu điểm của phương pháp này là cung cấp cho người thiết kế khả năng trừu tượng hóa hệ thống ở mức độ cao, đơn giản trong tính toán và sử dụng các biến quyết định nhị phân. Ứng dụng chính của FTA được sử dụng trong phân tích độ tin cậy của các hệ thống điện và mạng lưới truyền thông. Volkanovski và cộng sự [75] đã kết hợp FTA với phương pháp sử dụng mức cắt tối thiểu (MC) để đánh giá độ tin cậy của hệ thống mạng phục vụ phân phối điện năng. Sihombing và Torbol [65] đã phát triển kỹ thuật tính toán độ tin cậy của hệ thống bằng cách thực hiện song song hóa

dựa trên phương pháp FTA nhằm cải tiến và khắc phục tình trạng khi mà FTA gặp hạn chế đối với các bài toán cần giải quyết trên hệ thống phức tạp.

FTA là một phương pháp hữu hiệu để giải quyết vấn đề độ tin cậy, đặc biệt phù hợp với các hệ thống mạng lưới truyền tải điện năng khi mà trong hệ thống ngoài các thành phần chính, còn có những thành phần phụ tải cũng tham gia vào hoạt động của hệ thống chung. Tuy nhiên, ứng dụng của FTA đối với độ tin cậy của mạng còn khá hạn chế, các cây lỗi (Fault Tree) truyền thống không thể xử lý đối với các hệ thống phức tạp có sự biến động của các nút bên trong hệ, hay có sự phụ thuộc tuần tự của các nút với nhau.

### **1.3.5. Một số nhận xét**

Đánh giá độ tin cậy của các hệ thống và mạng lưới là một yếu tố quan trọng trong việc lập kế hoạch bảo trì và nâng cấp hệ thống một cách hiệu quả, đồng thời giúp xác định các điểm yếu trong hệ thống, từ đó thực hiện các biện pháp phòng ngừa để tránh sự cố và gián đoạn dịch vụ. Mỗi phương pháp đánh giá độ tin cậy có đặc điểm riêng và phù hợp với từng bài toán, yêu cầu khác nhau. MCS là phương pháp phổ biến được sử dụng kết hợp với các thuật toán tính toán độ tin cậy của mạng. Sự linh hoạt và đơn giản của MCS khiến nó phù hợp để kết hợp với các thuật toán khác nhau, tuy nhiên, thời gian mô phỏng dài, sự khác biệt của giải pháp và sự biến đổi kết quả đối với các mạng lớn đã hạn chế ứng dụng của nó. Trong khi đó, phương pháp chuỗi Markov lại phù hợp để đánh giá độ tin cậy của mạng đa trạng thái như các mạng lưới dòng điện ngẫu nhiên, hay mô hình hóa vấn đề nhiễu loạn do thời tiết có thể được xử lý tốt bằng các phương pháp chuỗi Markov. Tuy nhiên sự bùng nổ trạng thái của hệ thống là một mối quan tâm khi quyết định lựa chọn phương pháp này để đánh giá độ tin cậy.

Phương pháp BN phù hợp cho các mạng được kết nối chặt chẽ, do vậy các hiện tượng phụ thuộc lẫn nhau giữa các thành phần của mạng như cơ chế xếp tầng trong hệ thống mạng điện có thể được mô hình hóa tốt bằng BN. Bên cạnh đó, các nghiên cứu cho thấy phần lớn ứng dụng của BN tập trung vào đánh giá khả năng phục hồi và độ bền của hệ thống mạng. Trong khi phương pháp FTA dễ tính toán, hữu ích cho các mạng có cấu trúc đơn giản.

#### **1.4. Các phương pháp dự phòng nâng cao độ tin cậy hệ thống**

Dự phòng là một trong những phương pháp đơn giản và phổ biến nhất được sử dụng để cải thiện độ tin cậy và tính sẵn sàng của hệ thống [80]. Dự phòng đề cập đến việc các thành phần hoặc tài nguyên được bổ sung trong cơ sở hạ tầng hệ thống nhằm mục đích cải thiện độ tin cậy, tính sẵn sàng và khả năng phục hồi khi gặp sự cố hoặc hạn chế việc xảy ra các sự kiện không mong muốn trong quá trình vận hành. Trong hệ thống, các thành phần đóng vai trò là dự phòng được đặt ở chế độ không hoạt động (dự phòng) cho đến khi một thành phần hoạt động bị lỗi, khi đó thành phần dự phòng sẽ được sử dụng thay thế. Hiện nay có ba cơ chế dự phòng ở chế độ chờ thường được sử dụng trong thực tế gồm: chế độ chờ lạnh, chế độ chờ ấm và chế độ chờ nóng.

##### **1.4.1. Cơ chế dự phòng nóng**

Dự phòng nóng (hot standby) là một chiến lược dự phòng thường được áp dụng trong các hệ thống yêu cầu có độ sẵn sàng cao như: máy chủ truy cập Internet của một công ty hay hệ thống máy chủ phân giải tên miền DNS của nhà cung cấp dịch vụ ISP [69]. Đây là phương pháp mà thiết bị dự phòng hoạt động đồng thời song song với thành phần ban đầu, đảm bảo rằng một hệ thống, máy chủ hoặc tài nguyên được duy trì ở trạng thái sẵn sàng hoặc chế độ hoạt động ngay lập tức để thay thế cho hệ thống chính khi xảy ra sự cố [68]. Mục tiêu của dự phòng nóng là

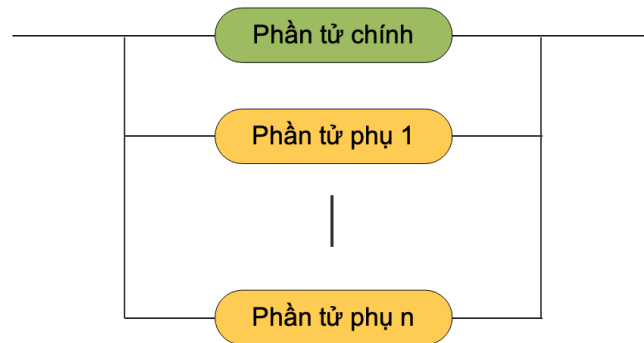
duy trì sự liên tục và không gián đoạn trong hoạt động của hệ thống khi phần tử chính gặp sự cố. Một số ưu điểm chính của cơ chế dự phòng nóng bao gồm [1, 64, 89]:

- **Nâng cao độ tin cậy cho hệ thống:** Ngay khi có sự cố xuất hiện hỏng xảy ra ở hệ thống chính, hệ thống dự phòng có thể tiếp tục hoạt động mà không mất thời gian chờ đợi lớn để chuyển đổi hay thực hiện khởi động.
- **Đảm bảo sự liên tục:** Dự phòng nóng giúp giảm thiểu thời gian ngừng hoạt động của hệ thống bằng cơ chế chuyển đổi tự động hoặc cần sự can thiệp tối thiểu từ phía người quản trị. Điều này giúp hệ thống duy trì hoạt động mà không ảnh hưởng đến người dùng hoặc dịch vụ.
- **Sự đồng bộ và đồng nhất:** Thiết bị được sử dụng trong dự phòng nóng thường được duy trì ở mức độ đồng bộ hoặc tương đương với thiết bị chính. Điều này giúp đảm bảo rằng dữ liệu và trạng thái của hệ thống được cập nhật liên tục và sẵn sàng để tiếp nhận các yêu cầu từ người dùng mà không gây ra mất mát dữ liệu.
- **Nâng cao hiệu suất hệ thống:** Cơ chế dự phòng nóng cho phép thực hiện việc bảo trì hệ thống thường xuyên hơn mà không gây ra sự gián đoạn trong hoạt động, từ đó tăng hiệu suất làm việc và giảm sự hỏng hóc trong quá trình vận hành hệ thống.

Tuy nhiên, cơ chế dự phòng này cũng có hạn chế nhất định như chi phí vận hành cao. Để triển khai một hệ thống dự phòng nóng thường đòi hỏi một mức độ đầu tư tài chính lớn hơn bình thường vì cần phải duy trì tài nguyên cho thiết bị dự phòng ở trạng thái hoạt động liên tục, điều này dẫn đến tăng chi phí vận hành và bảo trì hệ thống.

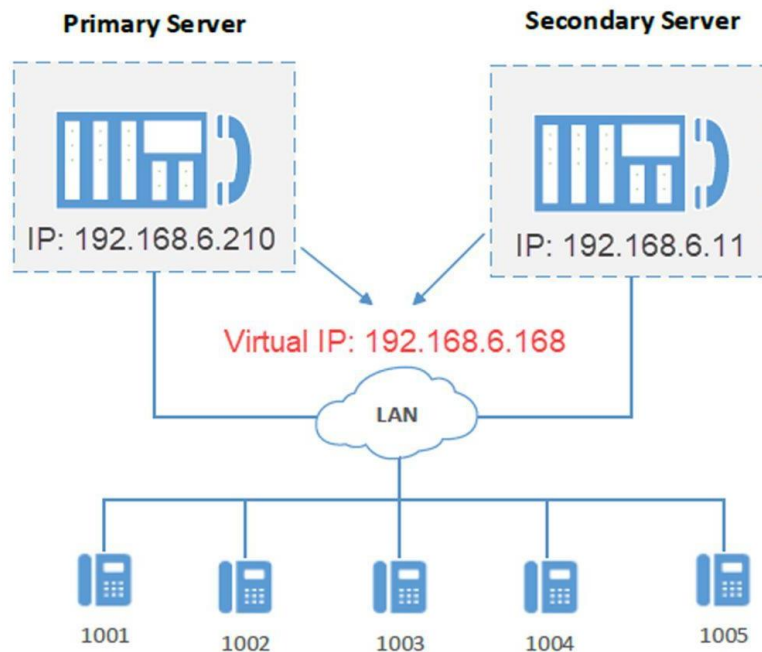
Trong thực tế, dự phòng nóng được ưu tiên áp dụng khi yêu cầu sự hoạt động liên tục ở cường độ cao và không ngừng của hệ thống, điều đó đồng nghĩa với việc chấp nhận chi phí đầu tư về tài chính sẽ cao hơn nhằm đảm bảo không sự cố nào gây ảnh hưởng đến hoạt động của hệ thống trong thời gian dài.

Giả sử hệ thống có 1 phần tử gốc và  $n-1$  phần tử dự phòng.



*Hình 1.8: Hệ thống các phần tử trong dự phòng nóng*

Trong hệ trên, các phần tử được sắp xếp làm việc theo mô hình song song, ở đây, phần tử dự phòng khi thay thế phần tử bị hỏng vẫn giữ nguyên chế độ tải trọng. Do vậy, độ tin cậy của các phần tử này không phụ thuộc vào thời điểm chuyển tiếp từ trạng thái dự phòng sang trạng thái làm việc. Chính vì đặc điểm này, nên hệ các phần tử hoạt động song song chỉ bị hỏng khi tất cả các phần tử trong hệ đều hỏng.



Hình 1.9: Cơ chế dự phòng nóng của tổng đài chăm sóc khách hàng

Mô hình hoạt động của tổng đài chăm sóc khách hàng trong thực tế với hai máy chủ hoạt động theo cơ chế dự phòng nóng, máy chủ phụ luôn ở trạng thái sẵn sàng hoạt động ngay khi máy chủ chính gặp sự cố. Mỗi máy chủ sẽ có địa chỉ IP trong mạng riêng, nhưng sẽ sử dụng chung địa chỉ IP ảo để các máy trạm trong mạng kết nối tới.

Tuy nhiên, do cơ chế làm việc của hệ thống song song, nên các phần tử luôn ở trạng thái sẵn sàng, đây là một hạn chế khi áp dụng cơ chế dự phòng này trong thực tế, vì chi phí đầu tư cho hệ thống sẽ tăng đáng kể tùy thuộc số lượng phần tử dự phòng được trang bị.

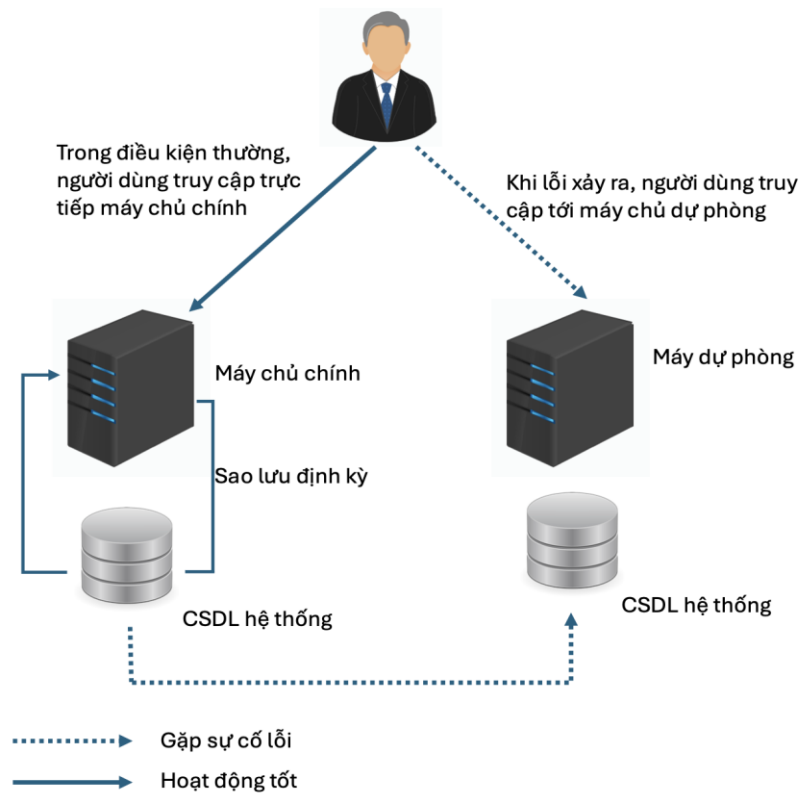
#### 1.4.2. Cơ chế dự phòng lạnh

Cơ chế dự phòng lạnh (Cold Standby) là một chiến lược dự phòng tiết kiệm hơn so với cơ chế dự phòng nóng về mặt tài chính do các tài nguyên phục vụ dự phòng được duy trì ở trạng thái mặc định là nghỉ (tức không sẵn sàng) cho đến khi

cần thiết phải thay thế hệ thống hoặc thiết bị chính khi có sự cố xảy ra [1, 26]. Một số đặc điểm chính của dự phòng lạnh bao gồm:

- Thời gian trễ: Trạng thái không hoạt động của hệ thống dự phòng lạnh yêu cầu một thời gian nhất định để tiến hành khởi động và sẵn sàng khi cần kích hoạt để thay thế cho hệ thống chính khi gặp sự cố hoặc được sử dụng trong việc bảo trì thường xuyên. Điều này có thể mất thời gian đáng kể hơn so với cơ chế dự phòng nóng, do chuyển đổi từ trạng thái nghỉ sang vận hành.
- Tiết kiệm chi phí: Dự phòng lạnh thường tiết kiệm chi phí hơn so với dự phòng nóng, do cơ chế này không yêu cầu duy trì tài nguyên dự phòng ở trạng thái hoạt động liên tục, giúp giảm thiểu các chi phí liên quan đến việc duy trì và vận hành.
- Dư thừa tài nguyên: Nếu hệ thống không sử dụng đến phần tử dự phòng, thì tài nguyên phục vụ cho dự phòng lạnh được coi như gây ra sự dư thừa và không tận dụng hiệu quả tài nguyên hệ thống.

Hệ thống dự phòng lạnh có thể hữu ích trong một số trường hợp khi hệ thống vận hành không yêu cầu dữ liệu phải được cập nhật thường xuyên và khi sự cố xảy ra thì thời gian ngưng trệ là có thể chấp nhận được để khôi phục hệ thống hoạt động trở lại.



Hình 1.10: Cơ chế dự phòng lạnh cho hệ thống máy chủ

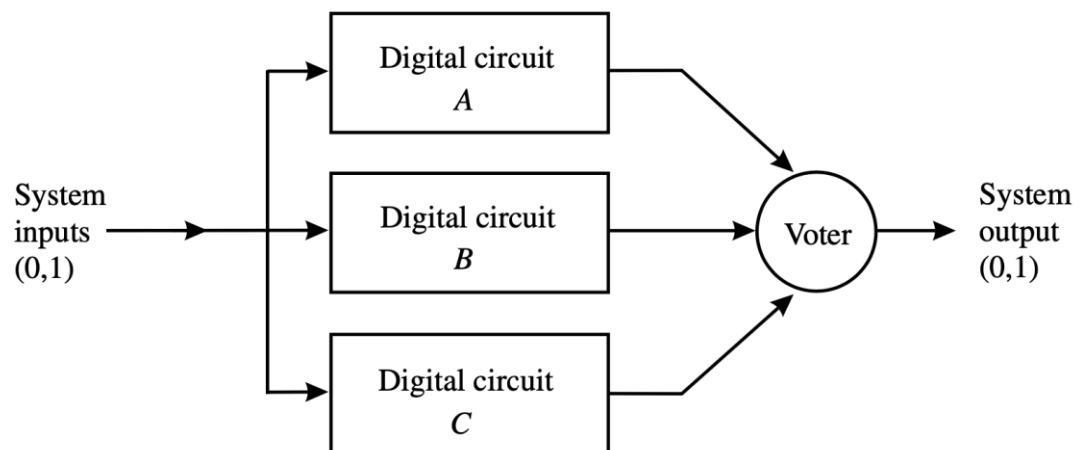
Mô hình hoạt động của hệ thống máy chủ dịch vụ áp dụng cơ chế dự phòng lạnh trong điều kiện bình thường, người dùng được cấp quyền truy cập tới hệ thống máy chủ chính. Trong trường hợp có sự cố xảy ra đối với máy chủ chính, kết nối của người dùng sẽ được chuyển sang cho máy chủ dự phòng. Tuy nhiên hệ thống dự phòng này không phải luôn hoạt động, người quản trị sẽ cần thời gian cho việc bật thiết bị và khôi phục cơ sở dữ liệu dự phòng trên hệ thống này trước khi cho phép toàn bộ hệ thống có thể vận hành như bình thường. Hệ thống này có thể được triển khai trong mô hình hoạt động của máy chủ xác thực miền sử dụng Active Directory trong hệ thống mạng nội bộ của công ty hoặc doanh nghiệp vừa và nhỏ.

### 1.4.3. Cơ chế dự phòng ấm

Dự phòng ấm (Warm Standby) là một phương pháp dự phòng trong đó một hệ thống phụ được duy trì ở trạng thái chờ, hoạt động một phần và sẵn sàng tiếp quản khi hệ thống chính gặp sự cố [34]. Hệ thống phụ không xử lý dữ liệu chính nhưng được cập nhật định kỳ để đảm bảo dữ liệu và trạng thái gần nhất với hệ thống chính. Khi cần, hệ thống phụ có thể khởi động nhanh chóng để đảm bảo tính liên tục của dịch vụ. Dự phòng ấm là một phương pháp dự phòng hiệu quả, cung cấp khả năng khôi phục nhanh chóng và đảm bảo tính liên tục của dịch vụ trong nhiều ứng dụng thực tiễn [26, 34, 78, 80, 87]. Nó cân bằng giữa chi phí và tính sẵn sàng, làm cho nó trở thành một lựa chọn lý tưởng cho các hệ thống đòi hỏi độ tin cậy cao như: hệ thống dịch vụ DNS [15].

#### 1.4.4. Cơ chế dự phòng kiểu chấp (dự phòng theo cơ chế bỏ phiếu)

Dự phòng kiểu chấp hay còn gọi là dự phòng theo cơ chế bỏ phiếu được sử dụng từ lâu trong các hệ thống và có nhiều nghiên cứu về ứng dụng của cơ chế dự phòng này [19, 64]. Trong dự phòng kiểu chấp, các phần tử của hệ thống được nhân ba số lượng, khi đó hệ thống sẽ làm việc nếu ít nhất hai trong ba phần tử đó làm việc và cho ra kết quả giống nhau.



Hình 1.11: Phương pháp dự phòng theo cơ chế chấp ba

Dữ liệu đầu vào của hệ thống sẽ được xử lý đồng thời tại A, B, C và cho ra các kết quả độc lập, bộ phận đóng vai trò quyết định đến việc lựa chọn kết quả đó là Voter. Trong mô hình hoạt động ở Hình 1.11, hệ thống sẽ so sánh kết quả làm việc của ba phần tử nếu có ít nhất hai phần tử trả về kết quả giống nhau thì lúc đó hệ thống sẽ đưa ra kết quả đầu ra.

#### **1.4.5. Một số nhận xét**

Để tận dụng những ưu điểm của mỗi cơ chế dự phòng, trong nhiều nghiên cứu gần đây, các nhà khoa học đã kết hợp các cơ chế lại với nhau khi tiến hành đánh giá độ tin cậy cho hệ thống. Trong nghiên cứu của Abouei và công sự [1], cơ chế dự phòng lạnh và dự phòng chủ động đã được kết hợp nhằm tối ưu hóa độ tin cậy của hệ thống. Sự kết hợp hai cơ chế dự phòng này trong cùng hệ thống sẽ cần phải xác định cụ thể số lượng phần tử hoạt động chủ động và số lượng phần tử đóng vai trò dự phòng ở mỗi hệ thống con trong hệ thống chính. Việc xác định này tương đối phức tạp để có thể đạt được giá trị tối đa độ tin cậy của hệ thống cần xây dựng. Nghiên cứu của Gao [26] đã thực hiện kết hợp ba cơ chế dự phòng gồm: dự phòng chủ động, dự phòng ấm và lạnh trong hệ thống, từ đó thực hiện phân tích độ tin cậy của hệ thống và đánh giá mức độ ảnh hưởng của lỗi xảy ra trên các phần tử trong hệ thống.

Việc kết hợp các cơ chế dự phòng khác nhau trong cùng một hệ thống có thể làm tăng độ phức tạp trong cấu hình, tuy nhiên với sự bổ sung các phương án dự phòng cũng sẽ tăng độ tin cậy cho hệ thống khi hoạt động. Bên cạnh đó, với cùng một cấu hình hệ thống, việc xác định phần tử nào sẽ được dự phòng (dự phòng có ưu tiên), số lượng phần tử sẽ được dự phòng, vị trí các phần tử dự phòng được bổ sung... cũng sẽ thay đổi cấu trúc, cũng như độ tin cậy của hệ thống. Chính vì vậy, cần thiết đưa ra một quy trình thực hiện đánh giá độ tin cậy của hệ thống

dựa trên các cấu hình dự phòng, từ đó xác định được phương án phù hợp với một số tiêu chí ban đầu mà các nhà thiết kế hệ thống đưa ra về độ tin cậy, thời gian hoạt động, khả năng vận hành ổn định sau khoảng thời gian nhất định.

### **1.5. Kết luận và vấn đề nghiên cứu**

Vấn đề tính độ tin cậy của các phần tử trong mạng đã và đang là một thách thức chủ yếu đối với các bài toán về độ tin cậy và nâng cao chất lượng dịch vụ trong hệ thống mạng. Do vậy, luận án này hướng đến việc nghiên cứu và cải thiện kỹ thuật tính toán độ tin cậy giữa hai phần tử đầu cuối trong mạng nhằm đảm bảo tính hiệu quả, chính xác và tối ưu hơn về thời gian tính toán.

Cụ thể là:

1- Nghiên cứu và phát triển phương pháp tính độ tin cậy dựa trên phương pháp SDP sử dụng kỹ thuật tính toán song song nhằm mục đích tăng tốc độ và cải thiện thời gian tính độ tin cậy giữa hai điểm đầu cuối trong mạng.

2- Nghiên cứu đánh giá độ tin cậy của các phương án dự phòng, từ đó đề xuất quy trình đảm bảo độ tin cậy cho hệ thống dựa trên các phương pháp dự phòng truyền thống, kết hợp với dự phòng tích cực.

## **CHƯƠNG 2. PHƯƠNG PHÁP ĐÁNH GIÁ VÀ CẢI THIỆN TÍNH ĐỘ TIN CẬY GIỮA HAI ĐIỂM ĐẦU CUỐI TRONG MẠNG**

Nội dung chương này tập trung về trình bày đóng góp khoa học trong nâng cao tốc độ tính toán độ tin cậy giữa hai phân tử bất kỳ trong mạng dựa trên phương pháp song song hóa. Kết quả nghiên cứu được thử nghiệm và so sánh với các phương pháp tính độ tin cậy khác để chứng minh sự ưu việt của thuật toán PNRE được đề xuất trong luận án.

### **2.1. Vấn đề đánh giá độ tin cậy giữa hai điểm đầu cuối trong mạng**

Bài toán đánh giá độ tin cậy giữa hai điểm đầu cuối trong mạng được coi là bài toán cơ bản và nền tảng cho các bài toán phức tạp hơn như: đánh giá độ tin cậy từ một điểm đến các điểm còn lại, hay đánh giá độ tin cậy của tất cả các điểm trong mạng. Để giải quyết bài toán cơ bản, có ba cách tiếp cận hiện được nghiên cứu chủ yếu là: Kỹ thuật SDP; Phương pháp xấp xỉ và Kỹ thuật lược đồ cây nhị phân BDD.

Tuy nhiên, kỹ thuật SDP [2, 30] được cho là có nhiều ứng dụng rộng rãi do việc triển khai cài đặt thực hiện đơn giản. Về cơ bản, kỹ thuật SDP xác định các đường dẫn nhỏ nhất cho biểu đồ tính độ tin cậy. Dựa trên độ tin cậy của các thành phần riêng lẻ và các đường dẫn cực tiểu nối giữa hai điểm đầu cuối có thể cho phép hệ thống hoạt động một cách an toàn, từ các thông số đó, thuật toán SDP sẽ tính độ tin cậy tương ứng để hệ thống hoạt động. Ý tưởng chính của SDP là tính tổng các thành phần rời rạc để xác định xác suất của tổng các tích, có thể được biểu thị bằng tổng xác suất của từng tích riêng lẻ. Khi sử dụng phương pháp SDP, việc đánh giá được thực hiện theo ba giai đoạn: 1) Tiến hành liệt kê tất cả các tập hợp đường dẫn tối thiểu (đường nhỏ hoặc lát cắt nhỏ); 2) Tính các tích rời rạc bằng

cách sử dụng thuật toán SDP; 3) Tính toán giá trị độ tin cậy của hệ thống dựa trên các chỉ số độ tin cậy của các thành phần riêng lẻ [30, 42].

Do kỹ thuật sẽ liệt kê toàn bộ các kết nối tồn tại giữa các nút trong một kiến trúc mạng, đây là công việc được thực hiện rời rạc có thể khá lớn và tốn nhiều thời gian thực hiện. Một giải pháp để cải tiến và tối ưu công việc tính toán này đó là thực hiện song song hóa các thao tác rời rạc và không có ảnh hưởng lẫn nhau.

Trong nghiên cứu tại [65], Sihombing đã đề xuất thuật toán mới sử dụng phương pháp tính toán song song sử dụng phần cứng GPU của hệ thống máy tính nhằm cải thiện tốc độ và độ chính xác của tính toán độ tin cậy trong hệ thống phức tạp dựa trên phương pháp phân tích cây lỗi (FTA). Thuật toán đã cho kết quả tốt hơn đáng kể so với phương pháp FTA ban đầu khi áp dụng trên các cây lớn có cấu trúc phức tạp với nhiều nút và sự kiện khác nhau. Nghiên cứu này cho thấy việc áp dụng các kỹ thuật tính toán mới và vận dụng sự phát triển của phần cứng để cải thiện hiệu năng đối với các phương pháp, thuật toán truyền thống.

Dựa trên cơ sở của thuật toán SDP, nhóm tác giả Hongjun trong nghiên cứu [16] đã đề xuất một thuật toán gọi là Fixed-Node Unconnected Subgraphs Algorithm (FUSA) dựa trên kiến thức về lý thuyết đồ thị và phương pháp liệt kê. Trong quá trình tính toán độ tin cậy của mạng, thuật toán này chỉ tìm kiếm các sơ đồ con mà không tạo ra tất cả các vectơ trạng thái mạng, điều này nhằm giảm đáng kể độ phức tạp tính toán. Ngoài ra, tập phần bù tương đối và tập cung bị loại trừ được giới thiệu trong thuật toán để tối ưu hóa việc tạo đồ thị con. Trên cơ sở đó, nhóm cũng đưa ra thuật toán có tên Quick Fixed-Node Unconnected Subgraphs Algorithm (QFUS). So với các thuật toán truyền thống, thì FUSA và QFUS có độ phức tạp về thời gian thấp hơn. Các ví dụ số cho thấy hiệu quả tính toán của QFUS

được cải thiện ít nhất 50 lần so với FUSA và QFUS hiệu quả hơn các thuật toán trực tiếp khác trong việc giải quyết vấn đề độ tin cậy của mạng.

Trong các nghiên cứu [82, 84], Yeh đã cải tiến phương pháp tính độ tin cậy SDP bằng cách tối ưu hóa việc xử lý các đường dẫn cực tiểu tìm được trong hệ thống mạng. Theo đó, từ danh sách các đường dẫn cực tiểu trong mạng ban đầu, tác giả không thực hiện so sánh đường dẫn  $A_i$  với các tất cả các  $A_j$  còn lại (với  $j < i$ ), mà chỉ tìm kiếm các đường dẫn ngắn hơn giữa hai đỉnh để so sánh và tính toán lại trong giải thuật của SDP. Độ phức tạp thời gian của phương pháp mới hiệu quả hơn các phương pháp hiện có, tuy nhiên tác giả cũng nhấn mạnh với các phương pháp tính toán theo cách tiếp cận SDP, thì việc so sánh nên được đánh giá dựa trên các tiêu chí như: Kích thước của kết quả thu được; Thời gian tính toán và Dạng của công thức thu được [82].

Trên thực tế, việc đánh giá trong một mạng lớn và phức tạp là một vấn đề đầy thách thức và tốn thời gian. Tuy nhiên, hầu hết các thuật toán hiện có đều được triển khai theo các bước tính toán tuần tự, điều này khiến tốc độ xử lý và thời gian thực thi không đáp ứng được trên các mạng phức tạp. Luận án thực hiện cải tiến và đưa ra phương pháp tính mới dựa trên kỹ thuật SDP, kết hợp với phương pháp tính toán song song. Kết quả thực nghiệm được áp dụng trên một số mô hình mạng và so sánh với một số thuật toán như LPC, SACNR.

## **2.2. Mô hình mạng và độ tin cậy của hai thiết bị đầu cuối**

### **2.2.1 Biểu diễn kết nối mạng trong lý thuyết đồ thị**

Trên thực tế, các hệ thống như mạng máy tính, hệ thống đường ống và lưới điện có thể được mô hình hóa thành mạng với các nút để có thể áp dụng các cơ sở lý thuyết về đồ thị, cạnh trong việc giải quyết vấn đề. Khi đó, có thể coi mạng  $G$  được biểu diễn dưới dạng đồ thị  $G(V, E)$  bao gồm  $n$  nút  $V$  và  $m$  cạnh  $E$ . Mỗi cạnh

chỉ có hai trạng thái: hoạt động hoặc lỗi. Một tập hợp con của các nút  $K$  ( $K \subseteq V$ ) (được gọi là thiết bị đầu cuối) là các phần tử vận hành. Một mạng sẽ hoạt động nếu có một hoặc nhiều đường dẫn hoạt động tới tất cả các thiết bị đầu cuối từ các thiết bị đầu cuối còn lại [6].

Trong bài toán hai thiết bị đầu cuối, dữ liệu được truyền từ nút nguồn  $s$  đến nút đích  $t$  và độ tin cậy của hai đầu cuối được xác định là sự hiện diện của một hoặc nhiều đường dẫn hoạt động giữa  $s$  và  $t$ .

Xét đồ thị vô hướng  $G$ , coi  $X_e$  là trạng thái của một cạnh  $e$  bất kỳ, nếu cạnh này ở trạng thái kết nối thì  $X_e = 1$ ; ngược lại  $X_e = 0$  có nghĩa là cạnh này xấu và có thể bị xóa khỏi đồ thị. Trạng thái của toàn bộ hệ thống được biểu diễn dưới dạng vectơ  $X$  bao gồm các giá trị  $X_e$  của các cạnh. Mỗi giá trị  $X_e$  chứa một giá trị ngẫu nhiên thể hiện cho trạng thái làm việc hoặc lỗi.

Gọi  $p_e$  và  $q_e$  lần lượt là xác suất hoạt động (làm việc) và không hoạt động (hỏng) của một cạnh  $e \subseteq E$ . Hay theo cách khác:

$p_e$  là xác suất của  $X_e = 1$  (ký hiệu là  $P_r(X_e = 1)$ ) và

$q_e$  là xác suất của  $X_e = 0$  (ký hiệu là  $P_r(X_e = 0) = 1 - p_e$ ).

Độ tin cậy  $R$  của mạng là xác suất hoạt động thành công, trường hợp độ tin cậy hai đầu cuối  $R_{st}$ , chúng ta có:

$$R_{st} = P(\text{các nút } s \text{ và } t \text{ được kết nối}) \quad (2.1)$$

Phương pháp đơn giản nhất để đánh giá độ tin cậy của cặp thiết bị đầu cuối trong mạng là liệt kê tất cả các kết hợp có thể có của các cạnh  $e$  trong đó mỗi cạnh là tốt hoặc xấu [64]. Mỗi sự kết hợp giữa trạng thái *hoạt động* và *không hoạt động* có thể được coi là một sự kiện. Các sự kiện này đều loại trừ lẫn nhau (hoặc rời rạc) và biểu thức độ tin cậy là sự kết hợp của các sự kiện này chứa đường dẫn giữa nút nguồn  $s$  và nút đích  $t$ .

Ta có thể viết như sau:

$$R_{st} = P(E_1 + E_2 + E_3 \dots) \quad (2.2)$$

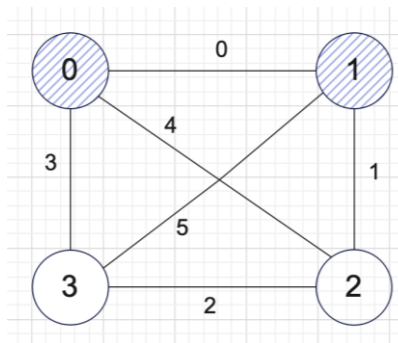
$E_i$  là các sự kiện độc lập với nhau hay đó chính là các cạnh tối thiểu chứa đường dẫn từ  $s$  tới  $t$ .

Mỗi sự kiện là độc lập với nhau nên:

$$R_{st} = P(E_1) + P(E_2) + P(E_3) + \dots \quad (2.3)$$

Trong đó:  $P(E_i)$  là xác suất xảy ra sự kiện  $E_i$  và phép toán (+) ở đây là hợp của hai đường dẫn tối thiểu.

Ví dụ với một đồ thị có bốn đỉnh như Hình 2.1:



Hình 2.1: Đồ thị với đỉnh nguồn “0” và đích “1”

Do đồ thị trên có 6 cạnh, nên có tổng số  $2^6$  sự kiện có thể xảy ra. Phương pháp liệt kê không gian sự kiện đòi hỏi nhiều công sức và thời gian mặc dù quy trình tính toán là rõ ràng, độ phức tạp của bài toán là  $2^e$ . Với giá trị  $e = 10$ , ta sẽ có 1.024 sự kiện và nếu giá trị  $e$  được tăng gấp đôi lên thì sẽ có hơn một triệu sự kiện.

Vì vậy với những mạng lớn có nhiều đỉnh và liên kết giữa các đỉnh tăng lên, thì việc xác định tất cả các sự kiện sẽ trở nên phức tạp và tốn kém về tài nguyên tính toán. Do đó, chúng ta cần tìm một phương pháp khác tốt hơn để tính toán với hệ thống có nhiều đỉnh và cạnh.

### 2.2.2. Sử dụng phương thức SDP trong tính xác suất tổng các thành phần

Sum of Disjoint Products (SDP) là phương pháp được sử dụng phổ biến trong việc tính xác suất hệ thống dựa trên việc phân tích thành tổng các xác suất của sự kiện riêng lẻ.

Mục đích là để giảm các thao tác và độ phức tạp trong việc phân tích độ tin cậy của mạng xuống dưới độ phức tạp  $2^e$ , có thể áp dụng phương pháp sử dụng các đường dẫn tối thiểu (bộ liên kết tối thiểu) của biểu đồ từ điểm nguồn tới điểm đích được đề cập trong các tài liệu [5, 6, 25, 64, 82]. Nếu có  $i$  tập tối thiểu giữa  $s$  và  $t$ , thì biểu thức tính độ tin cậy của hai điểm đầu cuối sẽ được viết lại như sau:

$$R_{st} = P(E_1 + E_2 + \dots + E_i) \quad (2.4)$$

Trong đó:  $P$  là xác suất tổng của các sự kiện;

$E_i$  là các cạnh hình thành lên đường dẫn từ điểm nguồn đến điểm đích.

Coi đỉnh nguồn là “0” và đỉnh đích là “1”, tức  $s = 0$  and  $t = 1$ . Ta có thể tìm được tập các cạnh tối thiểu giữa đỉnh nguồn và đích như sau:

**Bảng 2.1: Các đường dẫn tối thiểu từ  $s$  đến  $t$  trong Hình 2.1**

#	Tập tối thiểu	Đường dẫn $s \Rightarrow t$
1	$E_1 = \{0\}$	$0 \rightarrow 1$
2	$E_2 = \{4, 1\}$	$0 \rightarrow 2 \rightarrow 1$
3	$E_3 = \{4, 2, 5\}$	$0 \rightarrow 2 \rightarrow 3 \rightarrow 1$
4	$E_4 = \{3, 5\}$	$0 \rightarrow 3 \rightarrow 1$
5	$E_5 = \{3, 2, 1\}$	$0 \rightarrow 3 \rightarrow 2 \rightarrow 1$

Tập tối thiểu  $E_5 = \{3, 2, 1\}$  nghĩa là đi từ đỉnh nguồn đến đỉnh đích sẽ cần đi qua các cạnh “3”, “2”, và “1”. Trong khi đó, đường dẫn tối thiểu  $0 \rightarrow 3 \rightarrow 2 \rightarrow 1$  được hiểu rằng: đi từ đỉnh nguồn “0” đến đỉnh đích “1” sẽ lần lượt đi qua các đỉnh “3”, “2”, và “1”. Khi đó, độ tin cậy giữa đỉnh “0” và “1” có thể được tính như sau:

$$R_{01} = P(E_1 + E_2 + E_3 + E_4 + E_5) \quad (2.5)$$

Biểu thức trên là tổng hợp xác suất của các sự kiện, để có thể mở rộng phương trình  $R_{01}$  ở trên, có thể áp dụng công thức trong tài liệu [64]. Tuy nhiên

phương pháp này đòi hỏi nhiều nỗ lực tính toán và khó cài đặt. Vì vậy, nhiều phương pháp đã được đề xuất để chuyển đổi xác suất kết hợp của các sự kiện thành tổng các xác suất của mỗi sự kiện riêng lẻ (SDP). Đa phần các thuật toán dựa trên SDP để đánh giá độ tin cậy hai đầu cuối đều triển khai theo hướng tuần tự, nhưng có rất ít công bố liên quan đến các giải pháp song song hóa. Hiện nay ghi nhận có hai nghiên cứu về thuật toán SDP song song được đề cập trong tài liệu [20, 72].

### 2.3. Phương pháp tính toán độ tin cậy hai nút đầu cuối sử dụng thuật toán PNRE (Parallel Network Reliability Evaluation)

Trong kĩ thuật SDP có hai giai đoạn để thực hiện:

- Tìm kiếm tất cả các đường đi ngắn nhất giữa nút nguồn và nút đích.
- Tính toán độ tin cậy giữa hai nút.

#### 2.3.1 Tính toán xác suất của biểu thức logic dựa trên LPC

Hàm biểu thị xác suất hoạt động của hai nút (gọi là hàm xác suất) là tổng xác suất của các sự kiện riêng lẻ. Theo [60], hàm đại số logic liên kết trạng thái của từng phần tử trong hệ thống với trạng thái hệ thống được gọi là hàm xác suất vận hành của hệ thống  $y(x_1, x_2, \dots, x_n)$ .

Giả sử trạng thái của liên kết giữa hai nút  $\alpha, \beta$  là một trong hai trường hợp:

$$\begin{cases} \text{Làm việc } f_{\alpha\beta}(x_1, x_2, \dots, x_n) = 1 \\ \text{Lỗi } f_{\alpha\beta}(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (2.6)$$

Như vậy trạng thái của toàn hệ thống phụ thuộc vào trạng thái của từng phần tử  $x_1, x_2, \dots, x_n$ . Mỗi phần tử đều làm việc ( $x_k = 1$ ) hoặc lỗi ( $x_k = 0$ ).

Áp dụng công thức tính tổng xác suất các sự kiện riêng lẻ trong [60]:

$$P(A \vee B \vee C \vee D) = P(A) + P(\bar{A}B) + P(\bar{A}\bar{B}C) + P(\bar{A}\bar{B}\bar{C}D) \quad (2.7)$$

ta có công thức (2.7) sau đây:

$$P_{\alpha\beta} = P[f_{\alpha\beta} = 1] = P\left[\bigvee_{j=1}^m K_j = 1\right] = \sum_{j=1}^m P[K_j = 1] \quad (2.8)$$

Khi các sự kiện  $K_1, K_2, \dots, K_m$  loại trừ lẫn nhau, hàm đại số logic sẽ trở thành:

$$y(x_1, x_2, \dots, x_m) = \bigvee_{i=1}^n K_i (i \leq 2^m) \quad (2.9)$$

Công thức (2.8) có thể viết thành:

$$y(x_1, x_2, \dots, x_m) = K_1 \vee \bar{K}_1 K_2 \vee \bar{K}_1 \bar{K}_2 K_3 \vee \dots \vee \bar{K}_1 \bar{K}_2 \dots \bar{K}_{n-1} K_n \quad (2.10)$$

Và (2.10) tương đương với:

$$y(x_1, x_2, \dots, x_m) = K_1 \vee \bar{K}_1 (K_2 \vee \bar{K}_2 (\dots (K_{n-1} \vee \bar{K}_{n-1} K_n) \dots)) \quad (2.11)$$

Theo đó, công thức (2.11) có thể cài đặt được dễ dàng hơn công thức (2.9) vì có thể sử dụng kỹ thuật đệ quy để tính giá trị biểu thức  $(K_{i-1} \vee \bar{K}_{i-1} K_i)$  trong đó  $K_i$  chính là đường đi tối thiểu trong đồ thị.

### 2.3.2 Lưu đồ hoạt động của thuật toán PNRE

Lưu đồ thuật toán PNRE thực hiện các bước dựa trên thuật toán LPC trong Hình 2.2. Mục đích của thuật toán là biến đổi xác suất kết hợp của các sự kiện về dạng đơn giản hơn cho việc tính toán như thể hiện trong công thức (2.11).

Thuật toán PNRE được thực hiện thông qua năm bước chính bao gồm:

**Bước 1:** Sắp xếp ma trận  $W$  chứa tất cả các đường đi tối thiểu tìm được giữa hai điểm đầu cuối. Để giảm thiểu số lượng đường dẫn rời rạc được tạo ra, ta sẽ sắp xếp ma trận  $W$  theo thứ tự được giới thiệu trong nghiên cứu [12, 66].

**Bước 2:** Thuật toán chuyển đổi ma trận  $W$  thành vectơ  $C$  chứa các nút chung được lưu trữ trong tất cả các đường dẫn và ma trận  $K$  chứa các hàng sau khi loại trừ các nút chung.

**Bước 3:** Thuật toán khởi tạo ma trận  $R$  và biến  $i = n - 1$  ( $n$  là số hàng trong ma trận  $K$ ).

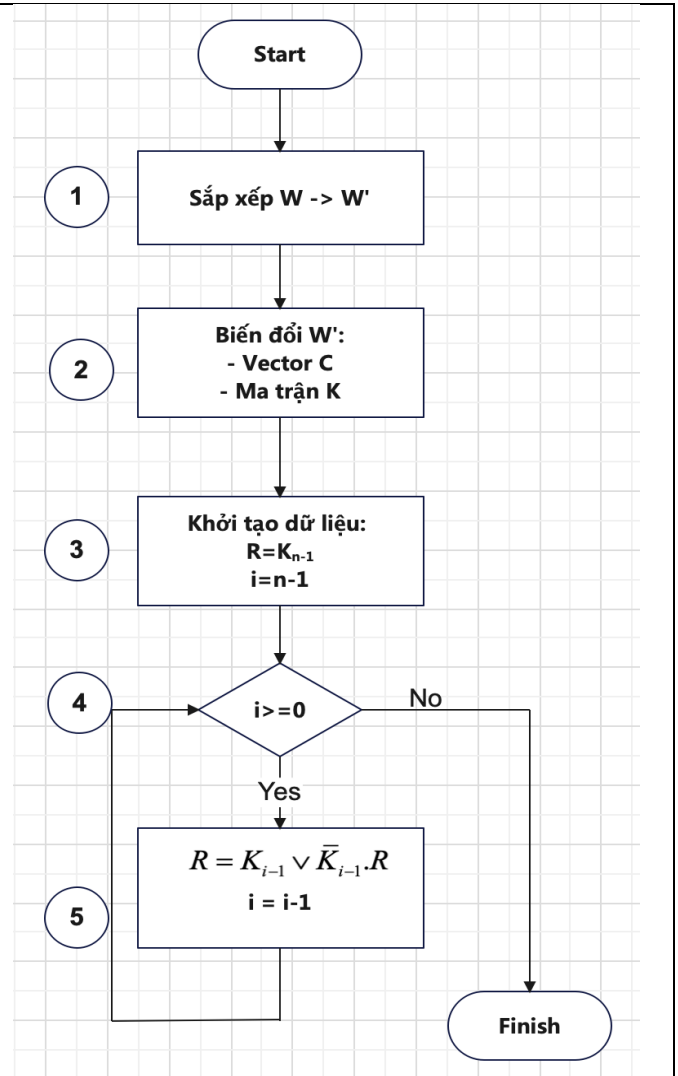
**Bước 4 & 5:** Thuật toán sẽ kiểm tra giá trị  $i \geq 0$  để thực hiện ba bước phụ:

- (1) Tìm ma trận nghịch đảo  $K_{i-1}$  (ký hiệu là  $\bar{K}_{i-1}$ );
- (2) Nhân ma trận  $\bar{K}_{i-1}$  và ma trận  $R$ ;
- (3) Thêm ma trận  $K_{i-1}$  vào kết quả.

Ngược lại, nếu giá trị  $i < 0$ : sẽ kết thúc giải thuật tính toán.

Chú thích các thành phần trong lưu đồ Hình 2.2:

- $W$  - là ma trận chứa tất cả các đường đi tối thiểu tìm được;
- $C$  - là vector chứa tất cả các nút chung nằm trên đường dẫn giữa hai điểm đầu và cuối;
- $K$  - là ma trận chứa  $K_i$  hàng trong Công thức (2.11);
- $N$  - là số hàng của ma trận  $K$ ;
- $R$  - là ma trận kết quả thu được sau khi tính toán.



Hình 2.2: Lưu đồ thuật toán PNRE

Kết quả thu được của thuật toán LPC cho ta một phương trình đại số như Công thức (2.10), từ đó sẽ thực hiện tính độ tin cậy giữa hai điểm đầu và cuối

trong topo mạng dựa trên giá trị xác suất từ đường đi tối thiểu đã tìm được trong các bước của thuật toán.

Từ kết quả thu được, ta có thể thực hiện cài đặt song song thao tác tính toán các biểu thức con  $t_i$  của  $K_i$ , như hình sau đây:

$$y(x_1, x_2, \dots, x_m) = K_1 \vee \bar{K}_1 K_2 \vee \bar{K}_1 \bar{K}_2 K_3 \vee \dots \vee \bar{K}_1 \bar{K}_2 \dots \bar{K}_{n-1} K_n$$

Hình 2.3: Tiến hành tính toán song song hóa các biểu thức trong Công thức (2.11)

Các thao tác tính toán  $t_i$  sẽ được thực hiện trong các luồng riêng và độc lập, do kết quả và thao tác tính toán không có sự liên quan lẫn nhau. Điều này sẽ giúp cho kết quả tính toán được thực hiện nhanh hơn rất nhiều, đặc biệt khi áp dụng trên các mô hình mạng phức tạp với nhiều nút, cạnh.

### 2.3.3. Đánh giá độ phức tạp thuật toán

Thuật toán PNRE được phát triển dựa trên cơ sở của thuật toán SDP, do vậy độ phức tạp của thuật toán PNRE về cơ bản là tương đồng. Tuy nhiên, trong thuật toán PNRE có sử dụng kỹ thuật tính toán song song để thực hiện tính các biểu thức từ công thức (2.11), do vậy có một số các điểm khác biệt như sau:

**Bảng 2.2: So sánh thuật toán PNRE và phương pháp SDP**

	<b>Phương pháp SDP ban đầu</b>	<b>Phương pháp PNRE</b>
Tìm đường đi	<ul style="list-style-type: none"> <li>- Độ phức tạp tính toán <math>O(n^2)</math></li> <li>- Việc lập trình để tìm các đường đi tối thiểu (MP) là khá thuận lợi trong việc sử dụng đệ quy.</li> <li>- Thao tác tính toán tuần tự, khó có thể thực hiện song song hóa các thao tác tính toán.</li> </ul>	<ul style="list-style-type: none"> <li>- Độ phức tạp tính toán <math>O(n^2)</math></li> <li>- Thuận lợi trong việc song song hóa thao tác tính từ biểu thức trực giao hóa nhân ma trận (Tại Bước 5 của lưu đồ Hình 2.3).</li> </ul>

Trực giao hóa		<ul style="list-style-type: none"> <li>- Sử dụng công thức tính Poresky và có thể triển khai song song hóa để tính toán.</li> <li>- Độ phức tạp của thao tác này là <math>O(n!)</math>.</li> </ul>
---------------	--	--

Như vậy, thuật toán PNRE không thực hiện cải thiện độ phức tạp của các thuật toán tương tự trước đây như SDP, mà thực hiện việc tăng tốc độ trong tính toán khi áp dụng kỹ thuật song song để thực hiện quá trình nhân các ma trận khi thực hiện trực giao hóa.

## 2.4 Cài đặt thuật toán PNRE

Trong lưu đồ thuật toán PNRE tại Hình 2.2, tại bước 5 ta cần thực hiện tính toán và trực giao hóa biểu thức logic  $K_i$ , để thực hiện được quá trình này, cần thiết phải xác định được các biểu thức logic thông qua một số kỹ thuật như cắt giảm, thuật toán Poresky...

### 2.4.1 Xác định trực giao hoá các toán tử logic

Giai đoạn trực giao hoá là giai đoạn phức tạp nhất của quá trình tính toán độ tin cậy của hệ thống. Nội dung mục này sẽ trình bày các phương pháp trực giao hoá, và các thuật toán được sử dụng để trực giao hoá các biểu thức logic.

Vị từ 1 ngôi (ký hiệu là  $f(x)$ ) được gọi là trực giao nhau khi và chỉ khi kết quả của chúng bằng 0. Tuy nhiên nếu vị từ mà ta xét không phải là vị từ 1 ngôi mà là vị từ  $n$  ngôi (ký hiệu là  $f(x_1, x_2, \dots, x_n)$ ) thì chúng được gọi là trực giao khi và chỉ khi các phân tử của chúng là trực giao và cho kết quả bằng 0. Vậy phương pháp trực giao chính là phương pháp biến đổi logic sao cho kết quả của các toán tử đều bằng 0.

**a. Phương pháp giảm thiểu các hàm đại số logic đối với các hình thức trực giao và trực giao không lặp.**

a) Thuật toán cắt giảm

Thuật toán này dựa trên việc triển khai các hàm logic, cho phép chúng ta thu được các hàm logic đã trực giao, không lặp lại và trong một số trường hợp đặc biệt, có thể biểu diễn dưới dạng chuẩn tắc tuyến trực giao. Cụ thể, thuật toán này hoạt động như sau:

1. Thực hiện xét hàm  $f(x_n)$  và thống kê số lần xuất hiện của mỗi biến  $x_i$  ( $i=1,2, \dots n$ ) dưới dạng sau:

$$M = \{m_1, m_2, \dots m_n\}$$

2. M được sắp xếp tăng dần khi đó  $m_n$  là tần số xuất hiện nhiều nhất tương ứng với biến  $x_i$  nào đó.

3. Đối với mỗi biến  $x_i$  ta thực hiện phép tách sau:

$$f(x_1, \dots, x_i, \dots x_n) = \bar{x}_i f(x_1, \dots, 0, \dots, x_n) \vee x_i f(x_1, \dots, 1, \dots, x_n) = \bar{x}_i f_i^{(0)} \vee x_i f_i^{(1)}.$$

4. Áp dụng các luật của đại số logic làm tối thiểu, đơn giản hoá các hàm  $f_i^{(0)}$  và  $f_i^{(1)}$ . Kết quả có thể nhận được sẽ thuộc 1 trong 4 trường hợp sau:

- Kết quả là hằng số 0 hoặc 1.
- Một hàm được đưa về dạng chuẩn tắc tuyến trực giao.
- Không có biến nào bị lặp lại trong hàm logic.
- Nếu có biến bị lặp lại trong hàm logic thì được loại bỏ bớt đi chỉ viết 1 lần.

5. Nếu kết quả hàm đã có sự kết hợp của cả trường hợp 1 và 2 thì việc chuyển hàm logic thành trực giao coi như đã xong. Nếu kết quả là trường hợp 3 thì tiếp tục áp dụng các luật biến đổi logic để hoàn tất việc trực giao hàm logic. Nếu kết quả là trường hợp 4 thì tiếp tục quay trở lại bước 1, thực hiện các phép biến đổi để đưa về dạng trực giao. Thủ tục này sẽ dừng vì số biến  $x_n$  là hữu hạn, khi đó ta được hàm logic đã trực giao.

Quá trình thực hiện thuật toán cắt giảm đối với các hình thức trực giao này tương đối mất thời gian và số lượng các biến tăng lên nhiều. Tuy nhiên thuật toán này lại có ưu điểm đó là vẫn nhanh hơn phương pháp tìm kiếm đầy đủ truyền thống, các hàm thu được bằng cách áp dụng các luật biến đổi logic sau khi áp dụng thuật toán hầu như sẽ không bị thay đổi gì nữa.

*b) Thuật toán trực giao hoá theo công thức của Poresky*

Thuật toán trực giao hoá theo công thức của Poresky [90] gồm các bước sau:

1. Hàm logic được viết dưới dạng chuẩn tắc tuyển như sau:

$$f(x_n) = K_1 \vee K_2 \vee \dots \vee K_j \vee \dots \vee K_R$$

Trong đó:  $K_j$  là mệnh đề sơ cấp có số thứ tự là  $j$ .

2. Tất cả các biến trong hàm logic khi được viết dưới dạng chuẩn tắc tuyển thì phải được sắp xếp theo thứ tự tăng dần, bắt đầu với bậc thấp nhất của biến.
3. Dạng chuẩn tắc tuyển được viết dưới dạng công thức Poresky như sau:

$$f(x_n) = K_1 \vee K_2 \vee \dots \vee K_j \vee \dots \vee K_R = K_1 \vee \overline{K_1} K_2 \vee \overline{K_1} \overline{K_2} K_3 \vee \dots \vee \overline{K_1} \dots \overline{K_{R-1}} K_R$$

4. Khai triển công thức ta được dạng trực giao không lặp của hàm ban đầu.

Ta thấy rằng việc sử dụng thuật toán đưa hàm logic về dạng chuẩn tắc tuyển đã trực giao này khá phức tạp trong việc triển khai các toán tử. Số lượng các biến khi khai triển là khá dài. Tuy nhiên đây chính là một cách để phát triển chương trình trực giao hoá của hàm logic trên máy tính.

Thuật toán trực giao hoá theo công thức Poresky có thể được sử dụng dưới hình thức sửa đổi một chút như sau:

1. Hàm logic được viết dưới dạng chuẩn tắc tuyển như sau:

$$f(x_n) = K_1 \vee K_2 \vee \dots \vee K_j \vee \dots \vee K_R$$

2. Từ dạng trực giao, sử dụng công thức De Morgan để tiếp tục biến đổi về dạng:

$$f(x_n) = \overline{(\overline{K_1} \overline{K_2} \dots \overline{K_R})}$$

3. Kết quả nhận được sau khi loại bỏ hết các biến lặp ta được dạng trực giao không lặp của hàm logic ban đầu. Dựa vào kết quả nhận được ta có thể dễ dàng tính các hàm xác suất.

*c) Thuật toán cắt giảm hàm đại số logic bằng cách đưa về dạng chuẩn tắc tuyến*

Như ta đã biết Orthogonal Disjunctive Normal Form (ODNF) – dạng chuẩn tắc tuyến hoàn toàn được coi như một dạng chuẩn tắc tuyến đã được trực giao hoá, vì vậy ta có thể sử dụng nó để cắt giảm các hàm logic. Thuật toán gồm các bước sau đây:

1. Đây là một hàm của dạng chuẩn tắc tuyến.
2. Tìm các phép hội sơ cấp xuất hiện trong dạng chuẩn tắc tuyến.
3. Tìm tất cả các tập hợp được bao phủ bởi mỗi phép hội sơ cấp
4. Khai triển tất cả các tập hợp, ta sẽ được hàm ban đầu trong dạng chuẩn tắc tuyến hoàn toàn.

Từ dạng chuẩn tắc tuyến hoàn toàn ta có thể dễ dàng tính được các hàm xác suất. Tuy nhiên nếu một hàm được viết ở dạng chuẩn tắc tuyến thì sẽ dài hơn khi viết ở dạng đã được trực giao, đây chính là một nhược điểm của thuật toán để đưa một hàm logic về dạng chuẩn tắc tuyến hoàn toàn. Nhưng việc triển khai một hàm logic về dạng chuẩn tắc tuyến hoàn toàn lại đơn giản hơn trong việc lập trình bằng máy tính.

### **b. Các quy tắc chuyển đổi hàm logic sang dạng xác suất trong dạng chuẩn tắc tuyến**

Bất kỳ một sự kiện phức tạp nào cũng có thể viết được dưới dạng hàm logic với các phép toán logic. Xác suất của hàm  $f(x_i)$  luôn bằng 1, được ký hiệu là  $P[f(x_i)]=1$ .

Khi hàm logic đã được chuyển đổi về dạng chuẩn tắc tuyến trực giao thì ta có thể tính xác suất của hàm một cách đơn giản theo quy tắc sau:

1. Các ký hiệu phép cộng và nhân logic được thay thế bởi dấu các phép cộng và phép nhân đại số.
2. Các ký hiệu  $x_i$  và  $\bar{x}_i$  được thay thế bằng các xác suất  $P_i$  và  $Q_i$  tương ứng  
Quá trình chuyển đổi về hàm xác suất được thực hiện theo quy tắc sau:
  1. Sử dụng công thức De Morgan để đưa về dạng chuẩn tắc hội, không chứa phép cộng logic nào.
  2. Các ký hiệu phép cộng và nhân logic được thay thế bởi dấu các phép cộng và phép nhân đại số.
  3. Các ký hiệu  $x_i$  và  $\bar{x}_i$  được thay thế bằng các xác suất  $P_i$  và  $Q_i$  tương ứng.
  4.  $\bar{P}_{(f=1)}$  được thay thế bởi  $1-\bar{P}_{(f=1)}$

### 2.4.2 Một số giải thuật được cài đặt trong thuật toán PNRE

#### a. Thuật toán đọc dữ liệu sơ đồ cấu trúc mạng từ tập tin:

Để có thể dễ dàng trong việc sử dụng và thử nghiệm trên các cấu trúc đồ thị khác nhau, chương trình đã cài đặt chức năng đọc dữ liệu cấu trúc mạng từ tập tin. Hàm đọc và phân tích dữ liệu từ định dạng XML để xác định số đỉnh, cạnh và khởi tạo ma trận C chứa tất cả cấu trúc của mạng.

Hàm chức năng readNetworkData đọc và phân tích dữ liệu
Đầu vào: Đường dẫn file XML
Đầu ra: Thiết lập biến lưu danh sách đỉnh, cạnh, nguồn, đích, xác suất cạnh
<pre>//Đọc dữ liệu từ file XML //Phân tích cấu trúc thẻ được định nghĩa trước đó để xác định các đỉnh, cạnh, điểm nguồn, điểm đích và xác suất của mỗi cạnh dataSet = ReadXML(srcFile) Khởi tạo NodeSet, LineSet tblNode = dataSet[node] for i ← 0 to tblNode.Rows.count do   row = tblNode.Rows[i]   node = {id, source, dest, probability} ← row   Thêm node vào NodeSet End for</pre>

```

tblLine = dataSet[line]
for i ← 0 to tblLine.Rows.count do
    row = tblLine.Rows[i]
    line = {type, begin, end, probability} ← row
    Thêm line vào LineSet
End for
Khởi tạo C_matrix(LineSet)

```

Sau khi đọc dữ liệu cấu trúc mạng và gán cho danh sách các nút, cạnh, hàm cũng gọi thực hiện chức năng khởi tạo ma trận C từ danh sách các cạnh được truyền vào.

```

1  <?xml version="1.0" standalone="yes"?>
2  <network>
3      <NodeSet>
4          <node id="0" src="true" dest="false" prob="1" />
5          <node id="1" src="false" dest="true" prob="1" />
6          <node id="2" src="false" dest="false" prob="1" />
7          <node id="3" src="false" dest="false" prob="1" />
8      </NodeSet>
9      <LineSet>
10         <line type="0" begin="0" end="1" prob="0.9" />
11         <line type="0" begin="1" end="2" prob="0.9" />
12         <line type="0" begin="3" end="2" prob="0.9" />
13         <line type="0" begin="3" end="0" prob="0.9" />
14         <line type="0" begin="0" end="2" prob="0.9" />
15         <line type="0" begin="1" end="3" prob="0.9" />
16     </LineSet>
17 </network>
18

```

Hình 2.4: Nội dung tập tin XML của cấu trúc mạng Topo1 trong Bảng 2.3

#### b. Hàm thực hiện chức năng khởi tạo ma trận C, K

Trong lưu đồ thuật toán, ma trận C và K được biến đổi từ ma trận W'. Trong đó C chứa những thành phần chung, còn K chứa các thành phần còn lại.

Giả sử từ ma trận W ban đầu:

$$W' = \begin{bmatrix} 0 & 1 & 5 & & & & \\ 0 & 1 & 2 & 5 & 8 & & \\ 0 & 1 & 3 & 7 & 9 & & \\ 0 & 1 & 2 & 3 & 6 & 8 & 9 \\ 0 & 1 & 2 & 3 & 5 & 6 & 7 \end{bmatrix}$$

Thực hiện biến đổi sang ma trận C và K như sau:

$$C = [0 \quad 1], \quad K = \begin{bmatrix} 5 & & & & & \\ 2 & 5 & 8 & & & \\ 3 & 7 & 9 & & & \\ 2 & 3 & 6 & 8 & 9 & \\ 2 & 3 & 5 & 6 & 7 & \end{bmatrix}$$

List createMT_K: Phương thức khởi tạo ma trận K
Đầu vào: ma trận W, tập nút phổ biến
Đầu ra: Ma trận K
<pre> Mt_MK ← rỗng for i ← 0 to mt_W.count do     Khởi tạo danh sách tạm list1     list2 = mt_W[i]     for j ← 0 to list2.count do         if not checkCurrPath(commNodeSet, list2[j])             list1.add(list2[j])         end if     end for     Mt_MK.add(list1) end for Return mt_MK </pre>

Trong phương thức trên có sử dụng hàm kiểm tra đường dẫn hiện tại có chứa đỉnh đang xét hay không, trong trường hợp không chứa thì sẽ kết nạp đỉnh đó và danh sách ma trận K. Biến *commNodeSet* được sử dụng để lưu trữ các đỉnh là thành phần chung có trong ma trận gốc W.

Từ danh sách các đỉnh trong đồ thị, cần tìm ra tất cả các đường dẫn tối thiểu nối từ đỉnh nguồn đến đỉnh đích. Việc tìm kiếm các đường đi này có thể được thực hiện bằng cách song song các tác vụ để giảm thời gian xử lý.

```
function findPaths(sourceNode, destNode): Phương thức tìm kiếm  
các đường đi từ đỉnh nguồn đến đỉnh đích
```

Đầu vào: đỉnh nguồn, đỉnh đích

Đầu ra: Các đường dẫn tồn tại giữa hai đỉnh

```
paths ← list rỗng  
taskList ← list rỗng  
  
for state ← 0 to numberOfNodes do  
    num = CM[sourceNode, state]  
    if state != sourceNode and num == 1  
        task = createTask(function(state)  
            currNode = state  
            allPaths ← list rỗng  
            tmpPath = new Path(sourceNode, currNode)  
            if currNode == destNode  
                allPaths.add(tmpPath)  
            else  
                for j ← 0 to numberOfNodes do  
                    num = CM[currNode, j]  
                    if j != currNode &&  
checkCurrPath(tmpPath, currNode) && num == 1)  
                        tmpPath.add(j)  
                        tmpPath.removeat(tmpPath.count - 1)  
                end for  
            end if  
            taskList.add(task)  
            startTask(task)  
        end if  
    end for  
  
for each task in taskList  
    for each path in getTaskResult(task)  
        addPathToList(paths, path)  
    end for  
end for
```

```
return paths
```

### Hàm thực hiện trực giao hóa biểu thức logic:

```
function orthogonalizeNEW(mt_K): Hàm trực giao hóa
```

Đầu vào: ma trận K

Đầu ra: Ma trận sau khi đã trực giao

```
taskList = createEmptyList()
MT = createEmptyList()

for state from 1 to count(MT_K) - 1
    task = createTask(function(state)
        index1 = state
        OM = createEmptyList()
        addToList(OM, MT_K[index1])
        for index2 from 0 to index1 - 1
            OM =mulMatrixMatrix(inverseMatrix(MT_K[index2]),OM)
        end for
        return OM
    end function, state)
    addToList(taskList, task)
    startTask(task)
end for

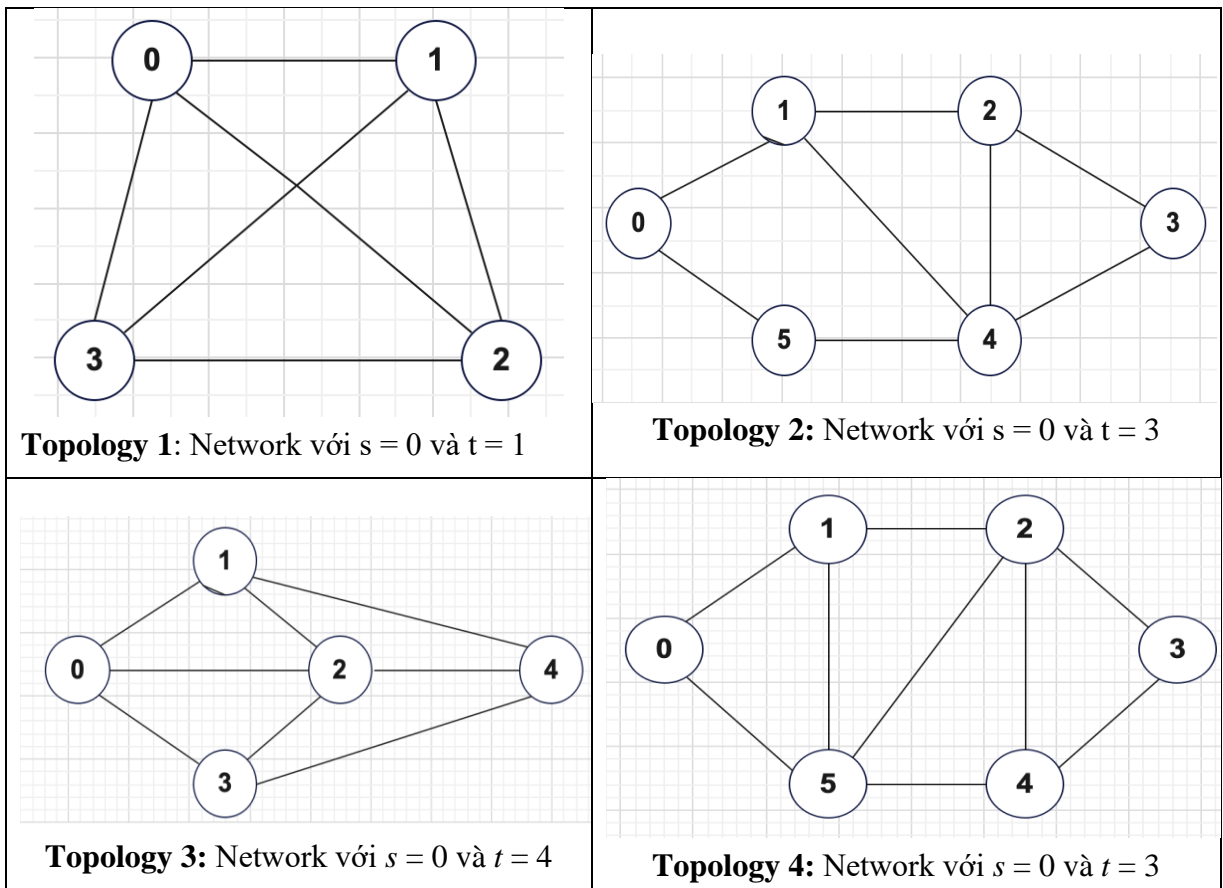
for each task in taskList
    for each intList in getTaskResult(task)
        addToList(MT, intList)
    end for
end for

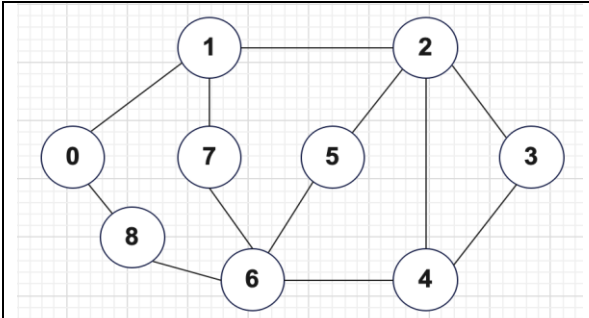
addToList(MT, MT_K[0])
sortMT(MT, RowComparer)
return MT
```

## 2.5 Thực nghiệm và so sánh phương pháp PNRE với LPC, SACNR

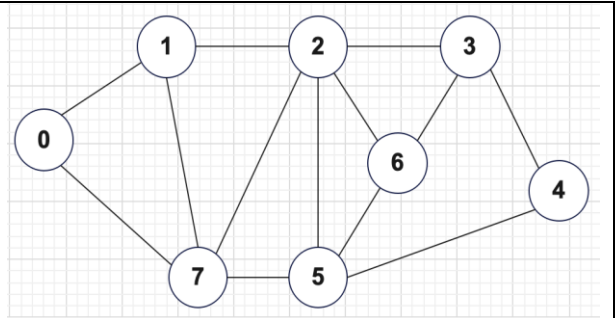
Để thực nghiệm và so sánh hiệu suất làm việc của phương pháp PNRE với các thuật toán khác như: LPC truyền thống và SACNR [85], đây là các thuật toán đều dựa trên phương pháp LPC. Luận án đã sử dụng một số mô hình mạng được tham khảo trong các tài liệu nghiên cứu (Topo mạng 1 [64], Topo mạng 2-9 [66]) của các tác giả có cùng lĩnh vực về đánh giá độ tin cậy trong hệ thống mạng, đây đều là các mô hình ngẫu nhiên với số lượng nút và cạnh khác nhau.

**Bảng 2.3: Các mô hình (topo) mạng được sử dụng trong thực nghiệm**

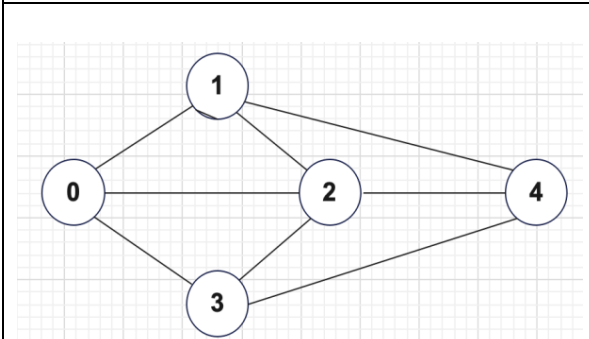




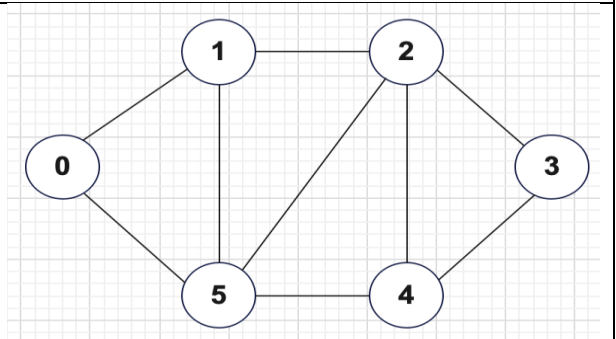
**Topology 5:** Network với  $s = 0$  và  $t = 3$



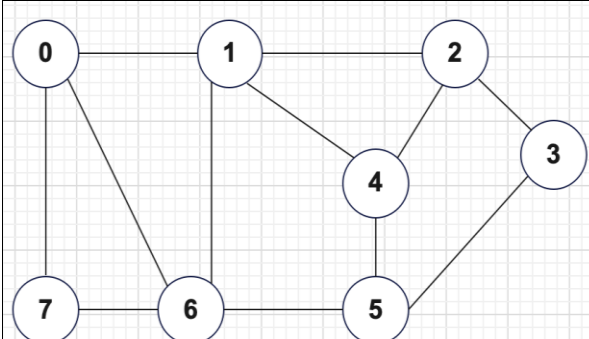
**Topology 6:** Network với  $s = 0$  và  $t = 4$



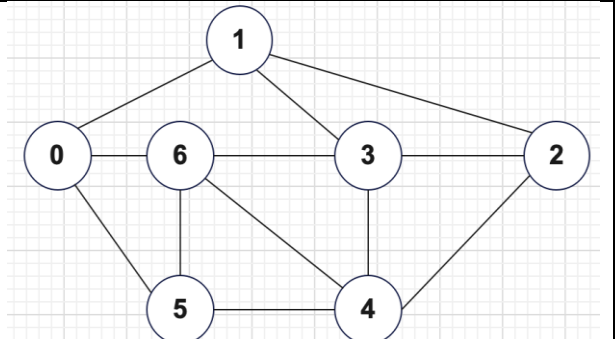
**Topology 3:** Network với  $s = 0$  và  $t = 4$



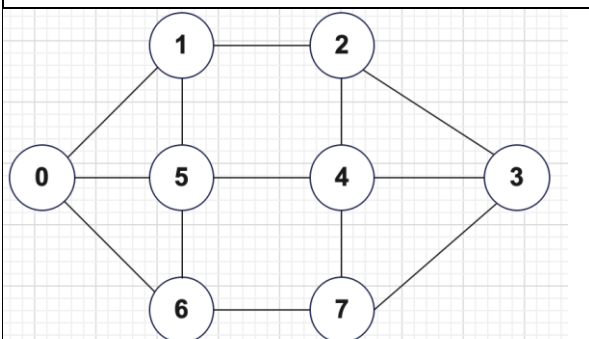
**Topology 4:** Network với  $s = 0$  và  $t = 3$



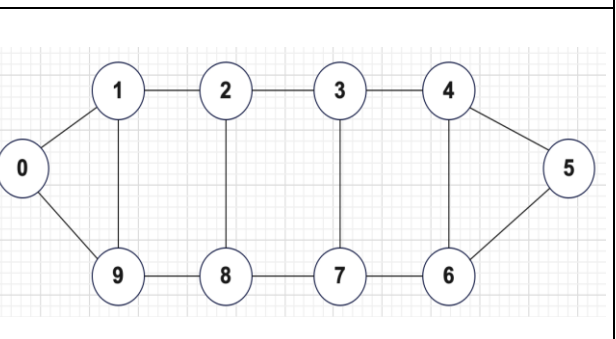
**Topology 7:** Network với  $s = 0$  và  $t = 3$



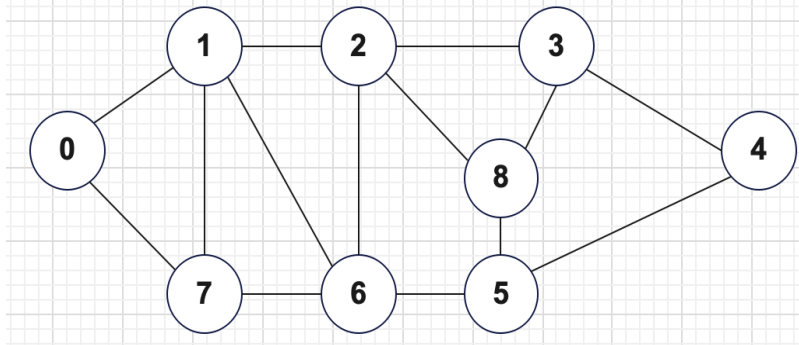
**Topology 8:** Network với  $s = 0$  và  $t = 2$



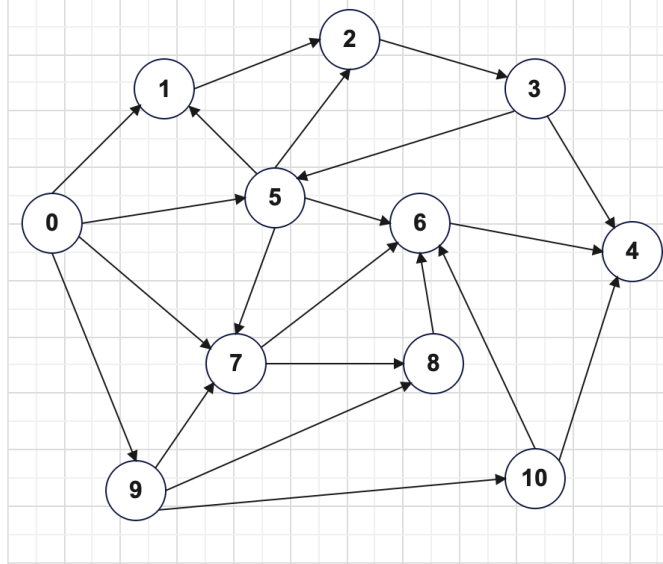
**Topology 9:** Network với  $s = 0$  và  $t = 3$



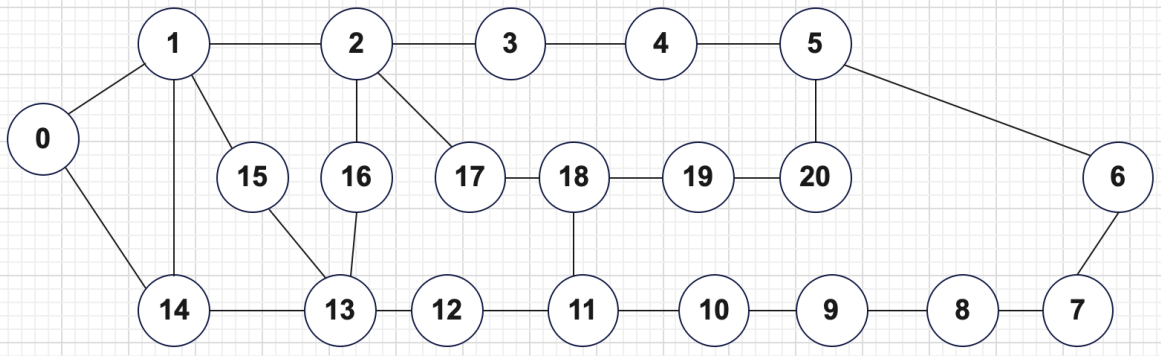
**Topology 10:** Network với  $s = 0$  và  $t = 5$



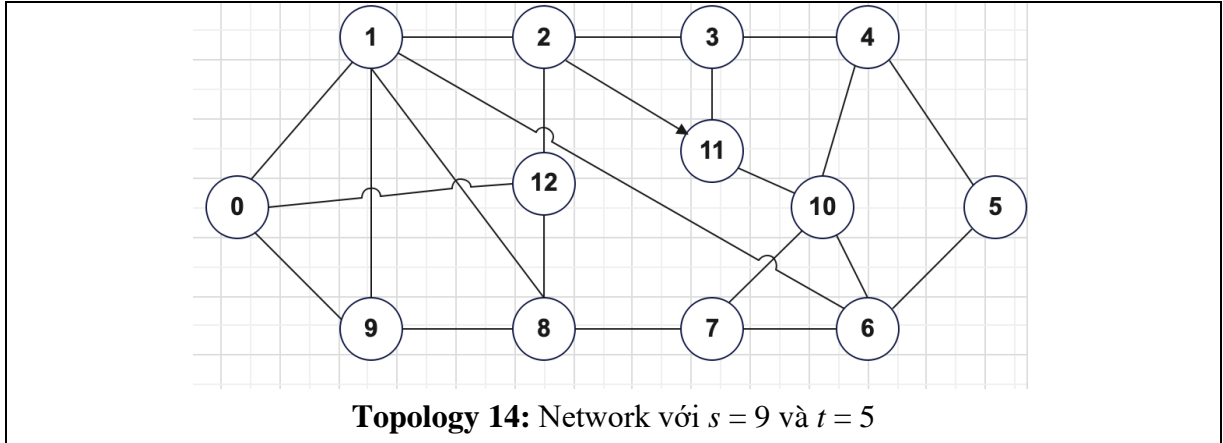
**Topology 11:** Network với  $s = 0$  và  $t = 4$



**Topology 12:** Network với  $s = 0$  và  $t = 4$



**Topology 13:** Network với  $s = 0$  và  $t = 6$



Chương trình cài đặt của ba thuật toán được thực thi trên cùng máy tính với bộ xử lý đa lõi có CPU Intel Core i5-8600, tốc độ 3,10 GHz gồm 6 lõi và hỗ trợ công nghệ siêu phân luồng, kết hợp với bộ nhớ 32 GB và hoạt động trên HĐH Windows 10 Pro (64-bit). Các thuật toán được cài đặt trên nền tảng Microsoft .NET Framework 4.6.1 SDK.

Giả sử các đỉnh trong các topo mạng đều có độ tin cậy lấy giá trị là 1, giá trị của các cạnh là 0,9. Thực hiện tính toán trên phần mềm cài đặt cho kết quả độ tin cậy của từng mạng như trong Bảng 2.4.

**Bảng 2.4: Thông số các mạng được dùng trong thực nghiệm**

Mạng	Số nodes	Số cạnh	Đường đi tối thiểu	Độ tin cậy
Topology 1	4	6	5	0,99785
Topology 2	6	8	7	0,96843
Topology 3	5	8	9	0,99763
Topology 4	6	9	13	0,97718
Topology 5	9	12	13	0,96485
Topology 6	8	12	24	0,97512
Topology 7	8	12	20	0,98407
Topology 8	7	12	25	0,99749
Topology 9	8	13	29	0,99622

Topology 10	10	14	32	0,94435
Topology 11	9	14	44	0,97415
Topology 12	11	21	18	0,99408
Topology 13	21	26	44	0,90458
Topology 14	13	22	281	0,98739

Tiến hành so sánh thời gian thực hiện giữa phương pháp PNRE với các thuật toán LPC và SACNR, sử dụng bộ đếm thời gian ngay trước và sau khi thực hiện các bước tính toán ở cả ba giải thuật tính toán, các giá trị đo đạc được trình bày trong Bảng 2.5 dưới đây:

**Bảng 2.5: So sánh thời gian thực hiện của PNRE với các thuật toán khác**

Network	SACNR (1)	LPC (2)	PNRE (3)	Speed up (2)/(3)	Speed up (1)/(3)
Topology 1	0,001	0,001	0,001	1	1
Topology 2	0,001	0,001	0,001	1	1
Topology 3	0,001	0,001	0,001	1	1
Topology 4	0,004	0,001	0,001	1	13
Topology 5	0,006	0,002	0,001	2	6
Topology 6	0,58	0,004	0,002	2	290
Topology 7	9,248	0,006	0,003	2	3.082,67
Topology 8	18,481	0,005	0,004	1,25	4.620,25
Topology 9	309,302	0,014	0,005	2,8	61.860,4
Topology 10	N/A	0.026	0.011	2.36	N/A
Topology 11	N/A	0.049	0.007	7	N/A
Topology 12	0.225	0.065	0.032	2.03	7.03
Topology 13	N/A	7.661	1.225	6.25	N/A
Topology 14	N/A	105.288	6.409	16.43	N/A

Kết quả đo đạc cho thấy với các topo mạng đơn giản có số lượng đỉnh ít như topo 1-2-3, thời gian thực thi của ba thuật toán gần như bằng nhau (trong khoảng 0,1giây). Thậm chí ở topo 4-5, sự khác biệt giữa LPC và PNRE thực sự không chênh lệch nhau đáng kể, nhưng thời gian thực thi của SACNR chậm hơn

so với PNRE tương ứng là 6 và 13 lần. Sự khác biệt bắt đầu tăng lên rõ ràng từ topo 6 trở đi, tốc độ của PNRE gấp 290x lần so với thuật toán SACNR. Đặc biệt ở các topo mạng 7-8-9, khi số lượng đỉnh và cạnh tăng lên kéo theo mức độ phức tạp của hình trạng mạng, điều này làm ảnh hưởng đến thời gian xác định đường đi từ đỉnh nguồn đến đỉnh đích và tốc độ tính độ tin cậy giữa hai đỉnh này. Giá trị cao nhất đo được là ở topo 9, khi tốc độ của thuật toán đề xuất PNRE nhanh gấp 61860 lần so với SACNR. Thậm chí trong một số trường hợp với topo phức tạp có số lượng đỉnh và cạnh lớn (trong topo mạng 9), thì thuật toán SACNR mất rất nhiều thời gian để có thể đưa ra được kết quả, với các topo 10, 11, 13, 14, thuật toán SACNR không trả về kết quả tính độ tin cậy sau khi thực hiện.

Như vậy với kết quả thực nghiệm, thuật toán PNRE cho thấy tính ưu việt trong khả năng tính toán độ tin cậy giữa hai điểm đầu cuối trong topo mạng cho trước. Thuật toán đã được cải tiến để tối ưu hơn khi áp dụng với các mô hình mạng phức tạp, điều mà một số thuật toán tính độ tin cậy truyền thống không thể thực hiện được hoặc thời gian xử lý là rất chậm.

## **2.6. Kết luận chương**

Nội dung trong chương này đã trình phương pháp đánh giá độ tin cậy trong hệ thống mạng và đề xuất phương pháp PNRE nhằm mục đích nâng cao độ tin cậy giữa hai điểm đầu cuối trong mạng. Kỹ thuật được sử dụng trong PNRE là song song hóa quá trình thực hiện tính toán dựa trên sức mạnh của các bộ vi xử lý máy tính. Kết quả thực nghiệm trên một số mô hình mạng cho thấy sự cải tiến về thời gian xử lý và tính toán của phương pháp mới so với các phương pháp truyền thống.

Dựa trên việc triển khai song song hóa thành công phương pháp tính độ tin cậy giữa hai điểm đầu cuối trong mạng, ta có thể mở rộng giải pháp tương tự với các bài toán với số nút lớn và topo mạng phức tạp. Ngoài ra, việc cải tiến hiệu quả

đánh giá độ tin cậy trong bài toán hai điểm đầu cuối có thể mở rộng và áp dụng vào các bài toán cùng lĩnh vực như: bài toán k điểm đầu cuối và bài toán tổng quát n điểm.

Nội dung của chương được tổng hợp và là kết quả của công trình đã được công bố tại [CT4].

## **CHƯƠNG 3. QUY TRÌNH ĐẢM BẢO ĐỘ TIN CẬY CHO HỆ THỐNG MÁY CHỦ DỰA TRÊN CƠ CHẾ DỰ PHÒNG**

Đảm bảo độ tin cậy đóng vai trò hết sức quan trọng trong việc duy trì hoạt động ổn định của một hệ thống thông tin, bên cạnh đó còn có ý nghĩa trong việc lập kế hoạch duy trì, vận hành cho hệ thống. Nội dung chương này sẽ trình bày một số cơ chế dự phòng nâng cao độ tin cậy cho hệ thống. Đồng thời đề xuất quy trình thực hiện nhằm đảm bảo độ tin cậy cho hệ thống dựa trên các cơ chế dự phòng, kết hợp với cấu trúc của hệ thống. Chương 3 được tổng hợp từ các công bố liên quan trong các công trình [CT1-CT3] và [CT5].

### **3.1. Cơ chế dự phòng nâng cao độ tin cậy cho hệ thống**

Trong bối cảnh phát triển và vận hành các hệ thống công nghệ thông tin ngày càng phức tạp và đa dạng, việc đảm bảo độ tin cậy và tính sẵn sàng của các dịch vụ là một yêu cầu cấp thiết. Độ tin cậy của hệ thống phản ánh khả năng duy trì hoạt động ổn định và không gặp sự cố trong suốt quá trình sử dụng. Có nhiều phương pháp để nâng cao độ tin cậy cho một hệ thống, tuy nhiên áp dụng các cơ chế dự phòng là phương án được sử dụng phổ biến nhất [26, 27].

Dự phòng là quá trình thiết kế và triển khai các biện pháp bảo vệ nhằm đảm bảo rằng hệ thống có thể tiếp tục hoạt động ngay cả khi một hoặc nhiều thành phần của nó gặp sự cố. Các cơ chế dự phòng không chỉ giúp giảm thiểu rủi ro về hỏng hóc mà còn tăng cường khả năng phục hồi và khôi phục nhanh chóng sau khi xảy ra sự cố.

Chính vì vậy, việc lựa chọn cơ chế dự phòng một cách hợp lý là rất quan trọng, vì nó không chỉ giúp bảo vệ hệ thống trước các sự cố bất ngờ mà còn đảm bảo tính liên tục của dịch vụ, từ đó tạo nên sự tin cậy và hài lòng từ phía người dùng. Khi một hệ thống có khả năng dự phòng tốt, doanh nghiệp có thể tránh được

những tổn thất nghiêm trọng về mặt tài chính và uy tín, đồng thời nâng cao hiệu quả hoạt động và khả năng cạnh tranh trên thị trường.

Nội dung chương này sẽ tập trung vào hai phương pháp dự phòng chính là: dự phòng song song và dự phòng tích cực.

### **3.1.1. Phương pháp dự phòng song song**

Phương pháp dự phòng song song là một kỹ thuật được sử dụng để cải thiện độ tin cậy và sẵn sàng của hệ thống, thường được áp dụng tại các hệ thống có yêu cầu cao về tính sẵn sàng và khả năng vận hành liên tục [64, 89]. Trong phương pháp này, nhiều thành phần hoặc hệ thống dự phòng được vận hành song song cùng với các thành phần chính. Khi một thành phần chính gặp sự cố, một trong các thành phần dự phòng có thể tiếp tục hoạt động mà không gây gián đoạn dịch vụ.

Trong phương pháp dự phòng song song, việc thiết lập, bố trí các thiết bị có vai trò dự phòng cho thiết bị chính có thể được chia làm 3 cơ chế khác nhau:

- Dự phòng nóng: Các hệ thống dự phòng hoạt động song song với hệ thống chính và luôn sẵn sàng để tiếp quản nhiệm vụ ngay lập tức nếu hệ thống chính gặp sự cố;
- Dự phòng ấm: Các hệ thống dự phòng được khởi động và sẵn sàng hoạt động, nhưng không hoạt động song song hoàn toàn với hệ thống chính. Hệ thống dự phòng ấm cần một khoảng thời gian ngắn để khởi động và tiếp quản khi hệ thống chính gặp sự cố.
- Dự phòng lạnh: Các hệ thống dự phòng không hoạt động cho đến khi hệ thống chính gặp sự cố. Khi xảy ra sự cố, hệ thống dự phòng lạnh cần được khởi động và có thể mất nhiều thời gian hơn để bắt đầu làm việc.

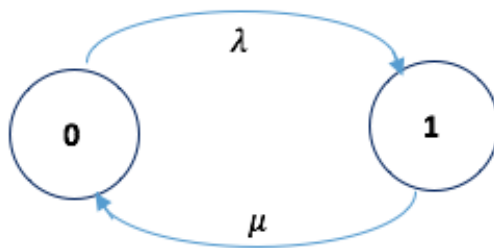
Do tính chất hoạt động khác nhau giữa ba cơ chế, nên dự phòng ảm được đánh giá là có tính ưu việt cao hơn hai loại còn lại về chi phí vận hành, khả năng sẵn sàng của hệ thống. Nghiên cứu của các tác giả trong [26, 34, 78, 80, 87] nhằm mục đích tối ưu hiệu suất hoạt động bằng cách thiết lập cơ chế làm việc hợp lý cho phần tử dự phòng dựa trên đánh giá các thông số dự phòng như: tỉ lệ sửa chữa, thời gian sửa, tỉ lệ và khả năng phục hồi của các phần tử dự phòng.

### 3.1.2. Dự phòng song song với phần tử có phục hồi

Trong cơ chế dự phòng song song này, các phần tử có khả năng phục hồi bằng cách sửa chữa hoặc thay mới sao cho phần tử đó lại có được những tính chất ban đầu.

Với hệ thống các phần tử có khả năng phục hồi, ngoài xác suất hỏng  $\lambda$ , phần tử còn có thêm giá trị gọi là xác suất phục hồi (ký hiệu là  $\mu$ ) [64, 71, 89]. Giá trị MTTR là thời gian trung bình cần thiết để sửa chữa hệ thống, MTTR thường được xác định theo tỉ lệ sửa chữa  $\mu$ , đó là số lượng dự kiến sửa chữa cho mỗi đơn vị thời gian. Thông thường tỉ lệ phục hồi lớn hơn nhiều so với tỉ lệ thất bại ( $\mu \gg \lambda$ ), MTTR được tính bằng:  $MTTR = \frac{1}{\mu}$  [71, 73].

Mô hình chuỗi Markov thường được sử dụng để mô hình hoá hệ thống với các phần tử có phục hồi, sơ đồ chuyển trạng thái sẽ có dạng:



Hình 3.1: Sơ đồ chuyển trạng thái Markov với các phần tử phục hồi

Nếu kí hiệu các xác suất chuyển trạng thái tương ứng là  $P_{00}(\Delta t)$ ,  $P_{01}(\Delta t)$ ,  $P_{10}(\Delta t)$  và  $P_{11}(\Delta t)$ . Theo mô hình Markov ta lập được ma trận xác suất chuyển trạng thái như sau:

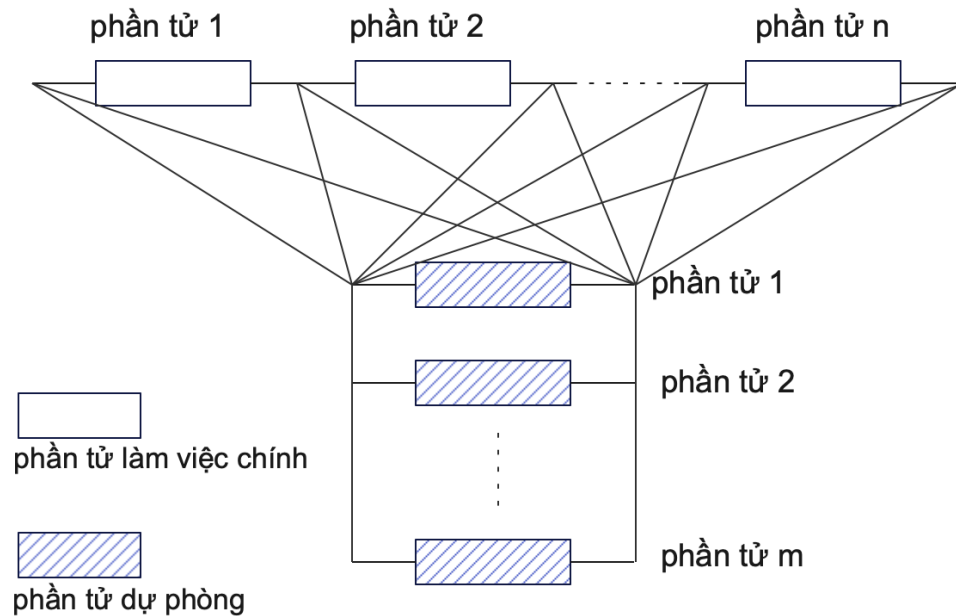
$$\begin{pmatrix} P_{00}(\Delta t) & P_{01}(\Delta t) \\ P_{10}(\Delta t) & P_{11}(\Delta t) \end{pmatrix}$$

Từ ma trận chuyển trạng thái có thể xây dựng được hệ phương trình vi phân đối với các xác suất trạng thái của hệ, giải hệ ta thu được hệ số sẵn sàng và hệ số không sẵn sàng của hệ thống.

### 3.1.3. Phương pháp dự phòng tích cực

Hệ thống dự phòng bảo vệ tích cực hay còn gọi là hệ thống dự phòng kiểu trượt, bao gồm có  $n$  phần tử cùng loại hoạt động theo kiểu nối tiếp. Nguyên tắc của hệ thống cần phải đảm bảo tất cả các phần tử đều làm việc, bất kỳ phần tử nào hỏng sẽ kéo theo hệ thống ngừng hoạt động. Chính vì vậy, để đảm bảo điều kiện làm việc cho hệ thống, các nhà thiết kế đã bố trí thêm  $m$  phần tử cùng loại ở vị trí dự phòng. Khi một phần tử bất kỳ trong số  $n$  phần tử làm việc bị hỏng thì bộ chuyển tiếp sẽ đưa một trong số  $m$  phần tử dự phòng vào thay thế [64, 89].

Trong sơ đồ hoạt động của cơ chế dự phòng kiểu trượt ở Hình 3.2, ta có thể thấy rằng: các phần tử dự phòng trong  $m$  phần tử có thể sẵn sàng thay thế bất cứ phần tử nào gặp sự cố trong  $n$  phần tử chính đang hoạt động. Với sự linh hoạt trong cơ chế hoạt động này, kiểu dự phòng trượt này có thể đem lại mức độ sẵn sàng cao cho hệ thống.



Hình 3.2: Cơ chế dự phòng tích cực

Tuy nhiên điểm hạn chế đó là các phần tử dự phòng và phần tử hoạt động chính cần đồng nhất về cấu hình, có thể sẵn sàng thay thế cho nhau khi cần thiết và hạn chế tối đa thời gian ngưng trệ khi thực hiện việc chuyển đổi. Điều này phụ thuộc vào bộ quản lý dự phòng trượt cần hoạt động ổn định và phát hiện chính xác phần tử hỏng để thực hiện thay thế kịp thời.

### 3.2. Bài toán đảm bảo độ tin cậy cho hệ thống máy chủ

#### 3.2.1. Phát biểu nội dung bài toán

Đảm bảo độ tin cậy cho hệ thống thông tin là công việc quan trọng của người thiết kế hệ thống ngay từ khâu đầu tiên. Công việc này chính là quá trình bảo vệ và duy trì tính toàn vẹn, an toàn và đáng tin cậy của thông tin trong một hệ thống. Điều này bao gồm một loạt các biện pháp và quy trình được thiết kế để đối phó với các rủi ro có thể ảnh hưởng đến thông tin, từ việc truy cập trái phép, sửa đổi không đáng kể, đánh cắp hoặc mất mát thông tin. Trong đó yếu tố đảm bảo

tính sẵn sàng và khả năng khôi phục hệ thống cần phải đáp ứng khi cần thiết và khả năng khôi phục nhanh chóng sau sự cố.

Trong nghiên cứu của Sadeghi tại [61], tác giả đã phân tích và đưa ra cách tiếp cận nhằm tối ưu hóa độ tin cậy cho hệ thống bằng việc kết hợp giữa dự phòng song song với cách bố trí, thiết đặt các phần tử theo kiểu tuần tự, nối tiếp. Kết quả cho thấy mô hình mới có thể tối đa hóa độ tin cậy chính xác của hệ thống tại thời điểm thực hiện nhiệm vụ, với khả năng sử dụng các thành phần không đồng nhất trong mỗi hệ thống con. Ví dụ minh họa chứng minh hiệu suất cao của mô hình trong việc xác định cấu hình để có thiết kế tối ưu và tăng độ tin cậy của hệ thống.

Nghiên cứu của Gao [26] về phân tích độ tin cậy và tối ưu hệ thống áp dụng cơ chế dự phòng tích cực với  $n$  phần tử hoạt động chính và  $w$  phần tử dự phòng ở chế độ dự phòng ấm và  $c$  phần tử dự phòng lạnh. Cấu trúc của hệ thống phức tạp với nhiều thành phần và cơ chế dự phòng khác nhau được xét đến trong nghiên cứu. Cùng hướng nghiên cứu, bài báo của nhóm Peiravi [56], Feizabadi [23] đề cập đến vấn đề nâng cao độ tin cậy trong hệ thống với nhiều cấu trúc phức tạp như tuần tự kết hợp với song song, cùng với cơ chế dự phòng tích cực và dự phòng truyền thống. Các nghiên cứu này hướng đến nội dung giải quyết mục tiêu của bài toán tối ưu hóa độ tin cậy bằng cách sử dụng phân bổ dự phòng của hệ thống.

Tuy nhiên các nghiên cứu ở trên chưa xét đến khía cạnh về việc đảm bảo độ tin cậy cho hệ thống. Giả sử trước khi triển khai một hệ thống trong thực tế, nhà thiết kế cần đưa ra mô hình về cấu trúc của hệ thống, bao gồm số lượng, các thành phần trong hệ thống, cách thức liên kết, kết nối với nhau (song song hoặc nối tiếp), phương án dự phòng cho các thành phần trong hệ thống, sử dụng cơ chế dự phòng nào. Ngoài ra, sau khi xây dựng được cấu trúc của hệ thống, dựa trên đó có thể tính được độ tin cậy của hệ thống, nhà thiết kế có thể hiệu chỉnh độ tin cậy **kỳ vọng**

lên cao hơn bằng cách thay đổi cấu trúc các thành phần trong hệ thống, hoặc bổ sung phần tử dự phòng tại một số vị trí quan trọng. Tuy nhiên, việc bổ sung phần tử dự phòng không phải luôn là phương án tối ưu nhất, vì có thể việc sử dụng dự phòng nhiều sẽ dẫn đến chi phí đầu tư cho dự án tăng lên vượt mức **khả dụng**.

Ngoài ra, vấn đề đảm bảo độ tin cậy cho hệ thống còn giúp nhà quản trị hệ thống có công cụ để kiểm tra, đánh giá độ tin cậy phục vụ cho khâu bảo trì, sửa chữa hệ thống theo định kỳ.

Xuất phát từ những mục tiêu như vậy, luận án đề xuất quy trình đảm bảo độ tin cậy dựa trên cấu trúc của hệ thống.

### **3.2.2. Đề xuất quy trình đảm bảo độ tin cậy cho hệ thống.**

Xuất phát từ yêu cầu cụ thể của hệ thống cần xây dựng, số lượng và vị trí các phần tử hoạt động chính trong mô hình mạng, từ đó đưa ra các phương án về vị trí đặt các phần tử dự phòng, thực hiện xây dựng công thức để tính độ tin cậy của hệ thống và tiến hành điều chỉnh dựa trên mô hình mạng để tìm ra phương án tối ưu nhất.

Các bước tiến hành cụ thể để xác định phương án dự phòng đảm bảo độ tin cậy cho hệ thống gồm:

#### **- Bước 1: Xác định yêu cầu của hệ thống**

Mục đích của bước này là xác định hiện trạng của hệ thống, cấu trúc hoạt động, sơ đồ khối logic và liên kết của các thành phần hoạt động bên trong hệ thống. Liên kết giữa các thành phần có thể là tuần tự hoặc song song hoặc đó có thể là thành phần phức hợp giữa các kiểu liên kết khác nhau. Ngoài ra cần xác định phương án dự án hiện tại mà hệ thống đang sử dụng. Với các thành phần hoạt động cần xác định đó là kiểu có khả năng phục hồi hay không.

Tiếp theo cần xác định yêu cầu cho hệ thống cần đạt được, như độ tin cậy kì vọng, thời gian cần hoạt động tiếp theo của hệ thống, số lượng thành phần có thể bổ sung, chi phí bổ sung.

**- Bước 2: Kiểm tra tiêu chí mới đã đáp ứng được trên hệ thống hiện tại**

Bằng cách thực hiện các bước tính toán dựa trên cấu trúc hiện tại của hệ thống để đưa ra giá trị tin cậy, từ đó so sánh các yêu cầu để quyết định thực hiện các bước tiếp theo.

**- Bước 3: Xác định cấu trúc dự phòng của hệ thống.**

Xây dựng cấu hình cho hệ thống với các phần tử kết nối, xác định mối liên hệ giữa các phần tử để từ đó biết được cấu trúc của hệ thống là song song hay nối tiếp. Phương án dự phòng có thể được lựa chọn như: dự phòng song song, dự phòng chập, dự phòng tích cực.

Đánh giá mức độ và vai trò của từng thành phần trong hệ thống để xác định độ ưu tiên của từng phần tử, điều này sẽ quyết định xây dựng phương án dự phòng xoay quanh phần tử, thành phần nào của hệ thống.

**- Bước 4: Cập nhật cấu trúc mới của hệ thống và tính lại độ tin cậy.**

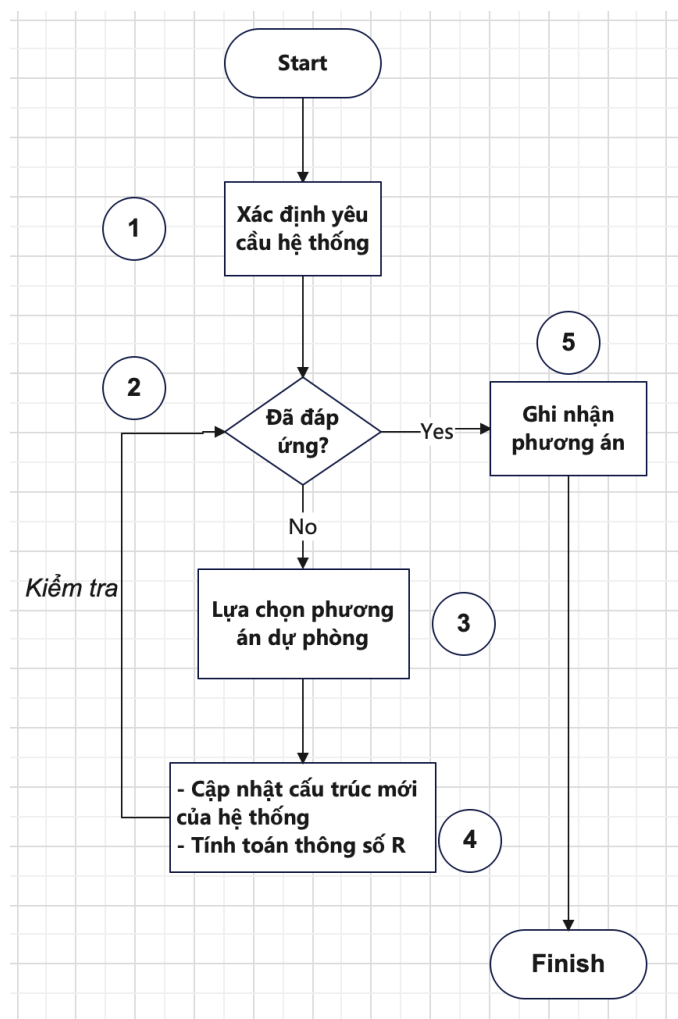
Dựa trên cấu trúc dự phòng mới đã xác định được từ Bước 3, ta tiến hành xây dựng công thức tính xác suất và độ tin cậy cho hệ thống mới, cần căn cứ vào phương pháp dự phòng được lựa chọn, **số lượng phần tử trong cấu trúc hệ thống ban đầu**, từ đó thiết lập công thức tính. Các phương pháp dự phòng như: dự phòng truyền thống, dự phòng chập hay dự phòng tích cực đều có đặc thù riêng trong việc xây dựng công thức tính toán.

Thực hiện đánh giá phương án dự phòng mới của hệ thống dựa trên độ tin cậy thu được. Tiến hành so sánh các giá trị thu được sau các mốc thời gian, từ đó quyết định phương án dự phòng mới có đạt mức kỳ vọng đặt ra hay không.

Nếu không đạt, có thể quay lại Bước 3 để thực hiện lại bằng cách thay đổi cấu hình dự phòng, vị trí đặt phân tử dự phòng hoặc thứ tự ưu tiên dự phòng trong hệ thống.

### - Bước 5: Ghi nhận phương án

Sau khi kết thúc Bước 4 đã có thể xác định được phương án dự phòng mới cho hệ thống đáp ứng theo tiêu chí đặt ra ban đầu về độ tin cậy, thời gian làm việc **kỳ vọng**, chi phí bổ sung cho đầu tư... Người quản trị cần thực hiện việc ghi nhận kết quả để tiến hành triển khai cấu trúc lại hệ thống, cũng như làm tài liệu phục vụ cho việc bảo trì, kiểm tra hệ thống định kỳ.



Hình 3.3: Quy trình đảm bảo độ tin cậy cho hệ thống

Quy trình này mang lại một cách tiếp cận tuần tự, từ việc xác định nhu cầu đến kiểm tra và cải tiến hệ thống, trong quá trình thực hiện, có các bước lặp lại để điều chỉnh và đánh giá nhằm lựa chọn phương án tốt nhất phù hợp với yêu cầu đề ra. Điểm mạnh của quy trình là tính tuần tự rõ ràng và tính nhất quán cao, giúp dễ theo dõi và điều chỉnh từng bước. Việc áp dụng quy trình trong tính toán độ tin cậy hệ thống có thể giúp cho các nhà quản trị, chuyên gia thiết kế hệ thống xác định được các phương án dự phòng khi xây dựng một hệ thống, cũng như điều chỉnh, thay đổi số lượng phần tử cần dự phòng để đạt được mục tiêu đặt ra. Tuy nhiên, cần lưu ý khi thực hiện:

- Phạm vi áp dụng: Quy trình này phù hợp với các hệ thống yêu cầu độ tin cậy cao như hệ thống mạng với số nút mạng vừa và nhỏ, do thực hiện và điều chỉnh các phương án sau mỗi bước.
- Giới hạn của quy trình: Quy trình có thể gặp khó khăn trong môi trường thay đổi nhanh chóng hoặc yêu cầu linh hoạt cao, do các bước có tính tuyến tính; Ngoài ra độ phức tạp của mô hình mạng và số lượng phần tử trong mạng cũng có thể ảnh hưởng tới việc xác định phương án dự phòng, cũng như xác lập công thức tính độ tin cậy.
- Việc xây dựng công thức tính độ tin cậy ở mỗi phương án trở nên phức tạp khi số nút dự phòng tăng lên, hoặc cấu hình dự phòng phức tạp khi kết hợp giữa các phương pháp.

### **3.3. Nâng cao độ tin cậy sử dụng phương pháp dự phòng song song**

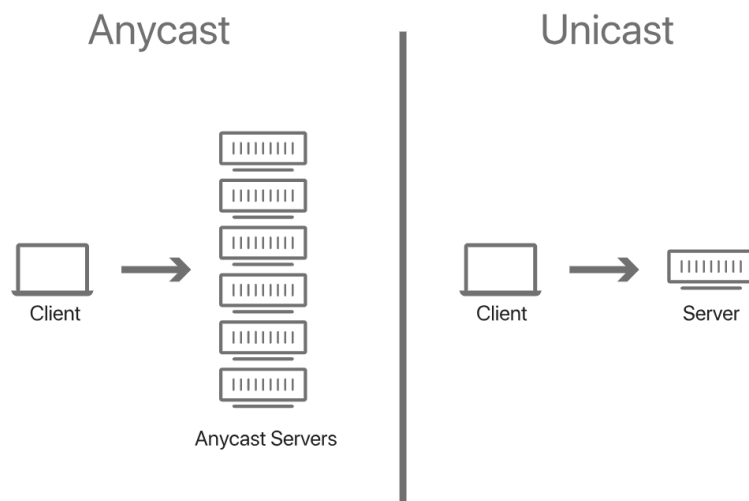
#### **3.3.1 Bài toán nâng cao độ tin cậy cho hệ thống máy chủ dịch vụ**

Trong mô hình hoạt động của hệ thống Internet ngày nay, dịch vụ DNS đóng vai trò hết sức quan trọng cho sự ổn định của toàn bộ mạng Internet toàn cầu.

Nhiệm vụ chính của DNS là chuyển đổi qua lại giữa tên miền với địa chỉ IP của các máy chủ tương ứng. Các máy chủ quản lý tên miền (còn gọi là DNS server) là nơi lưu trữ cơ sở dữ liệu phục vụ cho việc tìm kiếm và phân giải từ tên miền sang địa chỉ IP.

Với công nghệ định tuyến mạng Anycast, các nút mạng (như máy chủ, thiết bị mạng router) được sử dụng chung cùng một địa chỉ IP, cho phép triển khai một cách có hiệu quả và tối ưu tốc độ trả lời truy vấn tên miền trên các máy chủ DNS. Bên cạnh đó, các máy chủ DNS Anycast cũng nâng cao khả năng chịu tải truy cập cho hệ thống và mở rộng các nút mạng gần như không có giới hạn.

Trong quá trình truy vấn tên miền của các máy chủ trên mạng Internet, các yêu cầu từ máy client sẽ được định tuyến tới những máy chủ DNS Anycast gần nó nhất. Do đó, ứng dụng công nghệ Anycast cho các máy chủ DNS được cho là phát huy một loạt những thuận lợi: giảm độ trễ truy vấn, tăng độ tin cậy và khả năng sẵn sàng của hệ thống cũng như khả năng phục hồi sau các cuộc tấn công DDoS [40, 76]



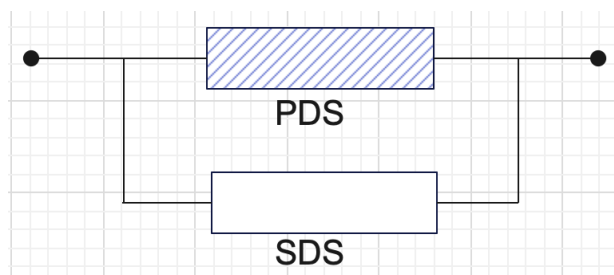
Hình 3.4: Mô hình hoạt động của hệ thống máy chủ DNS Anycast

Tuy nhiên, với sự bùng nổ Internet như hiện nay, bên cạnh số lượng người dùng Internet cũng như các thiết bị IP gia tăng, thì nhu cầu truy cập Internet và sử dụng các dịch vụ mạng càng tăng lên; Việc này đồng nghĩa với dịch vụ phân giải tên miền DNS sẽ phải hoạt động với cường độ lớn hơn. Mặc dù các máy chủ DNS Anycast được cấu hình tự động để phân phối các truy vấn của người dùng, cũng như chia tải giữa các máy chủ theo thuật toán định tuyến. Tuy nhiên việc quá tải cục bộ các DNS server là điều hoàn toàn có thể xảy ra, với số lượng truy vấn tăng lên đột biến khi hacker thực hiện các cuộc tấn công DDOS vào các máy chủ DNS, dịch vụ cung cấp cho người sử dụng có thể ngừng trệ [62, 67, 76].

Phương án đề xuất là sử dụng cơ chế dự phòng song song để bổ sung máy chủ dự phòng cho máy chủ đang hoạt động. Luận án thực hiện xét hai phương án: dự phòng một máy chủ và dự phòng hai máy chủ, từ đó đưa ra nhận định, đánh giá sự cải thiện về độ tin cậy giữa các phương án.

#### ***a. Phương án bổ sung một máy chủ dự phòng***

Giả sử hệ thống đang xét là máy chủ dịch vụ DNS gồm hệ hai máy chủ đồng nhất gồm: Primary DNS (PDS) và Secondary DNS Server (SDS). Theo nguyên lý song song, thì hệ thống hoạt động khi ít nhất có một máy chủ hoạt động. Các phần tử của hệ có thêm khả năng phục hồi sau khi gặp sự cố.



Hình 3.5: Mô hình hai máy chủ DNS Anycast hoạt động song song

#### ***a.1. Trường hợp phần tử không phục hồi***

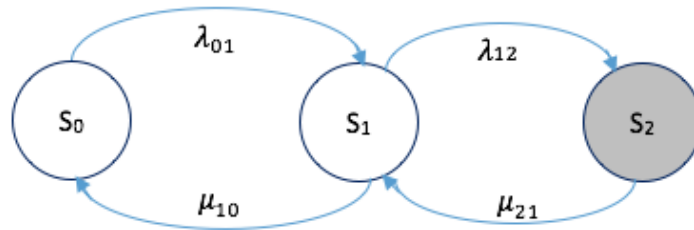
Theo [CT9], với hệ gồm hai phân tử dự phòng song song không phục hồi, ta có công thức xác định độ tin cậy phụ thuộc theo thời gian  $t$  như sau:

$$P_{\text{hệ}}(t) = p(t)^2 \quad (3.1)$$

với  $p(t) = e^{-\lambda t}$

### a.2. Trường hợp phân tử có phục hồi

Sử dụng chuỗi Markov, giả định các trạng thái hoạt động của hai máy chủ trong hệ thống như sau:



Hình 3.6: Sơ đồ chuyển trạng thái của hệ thống với hai phân tử

Hệ trên gồm ba trạng thái  $S_0$ ,  $S_1$ ,  $S_2$  tương ứng:

- Trạng thái  $S_0$ : Trường hợp cả hai phân tử cùng hoạt động đồng thời.
- Trạng thái  $S_1$ : Một phân tử ở trạng thái hoạt động tốt – phân tử còn lại rơi vào trạng thái hỏng.
- Trạng thái  $S_2$ : Cả hai phân tử cùng hỏng đồng thời.

Quá trình chuyển đổi trạng thái trong hệ xảy ra khi phân tử chuyển từ trạng thái làm việc sang trạng thái hỏng và ngược lại, kí hiệu  $\lambda$  và  $\mu$  tương ứng là tỉ lệ hỏng và phục hồi của phân tử.

Theo lý thuyết về chuỗi Markov, ta xây dựng được hệ phương trình vi phân của các trạng thái dịch chuyển trong hệ thống như sau:

$$(3.2)$$

$$\begin{cases} P_0'(t) = -\lambda_{01}P_0(t) + \mu_{10}P_1(t) \\ P_1'(t) = -(\mu_{10} + \lambda_{12})P_1(t) + \lambda_{01}P_0(t) + \mu_{21}P_2(t) \\ P_2'(t) = -\mu_{21}P_2(t) + \lambda_{12}P_1(t) \\ P_0(t) + P_1(t) + P_2(t) = 1 \end{cases}$$

Do các phần tử là đồng nhất nên xác suất hỏng  $\lambda$  và xác suất phục hồi  $\mu$  của mỗi phần tử là như nhau. Khi đó ta có thể coi  $\lambda_{01} = 2\lambda$  (do xác suất hỏng của một trong hai phần tử là như nhau).

Tương tự như vậy ta có:  $\lambda_{12} = \lambda$  và xác suất phục hồi tương ứng:  $\mu_{21} = 2\mu$  và  $\mu_{10} = \mu$ .

Hệ phương trình (3.2) trở thành:

$$\begin{cases} P_0'(t) = -2\lambda P_0(t) + \mu P_1(t) \\ P_1'(t) = -(\mu + \lambda)P_1(t) + 2\lambda P_0(t) + 2\mu P_2(t) \\ P_2'(t) = -2\mu P_2(t) + \lambda P_1(t) \\ P_0(t) + P_1(t) + P_2(t) = 1 \end{cases} \quad (3.3)$$

Để giải hệ phương trình vi phân (3.3) ở trên, giả sử tại thời điểm ban đầu  $t=0$ :  $P_0(0)=1$  và  $P_1(0)=P_2(0)=0$ , tức là cả hai phần tử đều ở trạng thái sẵn sàng hoạt động. Khi đó hệ (3.3) trở thành:

$$\begin{cases} -2\lambda P_0 + \mu P_1 = 0 \\ -(\mu + \lambda)P_1 + 2\lambda P_0 + 2\mu P_2 = 0 \\ \lambda P_1 - 2\mu P_2 = 0 \\ P_0 + P_1 + P_2 = 1 \end{cases} \quad (3.4)$$

Thực hiện giải hệ phương trình, ta thu được các giá trị:

$$P_0 = \frac{\mu^2}{(\lambda+\mu)^2}, P_1 = \frac{2\mu\lambda}{(\lambda+\mu)^2}, P_2 = \frac{\lambda^2}{(\lambda+\mu)^2}$$

Đặt giá trị  $\rho = \frac{\lambda}{\mu}$ , khi đó:  $P_0 = \frac{1}{(1+\rho)^2}$ ,  $P_1 = \frac{2\rho}{(1+\rho)^2}$ ,  $P_2 = \frac{\rho^2}{(1+\rho)^2}$

Hệ số sẵn sàng của hệ chính là:

$$K_r = P_0 + P_1 = \frac{1 + 2\rho}{(1 + \rho)^2} \quad (3.5)$$

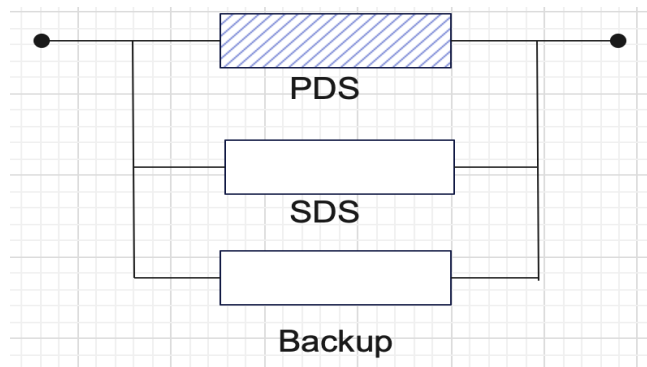
Trong đó:  $K_r$  là hệ số sẵn sàng của hệ.

$P_i$  là xác suất làm việc ở trạng thái  $i$ .

$\rho$  là biểu thức tương quan giữa xác suất hỏng và phục hồi.

### ***b. Phương án bổ sung hai máy chủ dự phòng***

Giả sử trong trường hợp hệ thống được trang bị thêm một phần tử dự phòng cho phần tử phụ, tức là khi này hệ sẽ bao gồm ba phần tử hoạt động song song theo Hình 3.7. Mô hình hệ thống với ba máy chủ dịch vụ DNS anycast đồng nhất được sắp xếp như sau:



Hình 3.7: Mô hình ba máy chủ DNS hoạt động có dự phòng

- Trong đó:
- PDS là máy chủ DNS chính;
  - SDS là máy chủ DNS dự phòng;
  - Backup là máy chủ dự phòng trong trường hợp SDS lỗi.

#### ***b.1. Trường hợp phần tử không phục hồi***

Theo [CT9], với hệ gồm ba phần tử dự phòng song song không phục hồi, ta có công thức xác định độ tin cậy phụ thuộc theo thời gian  $t$  như sau:

$$P_{\text{hệ}}(t) = p(t)^3 + 3\alpha(1 - p(t))p(t)^2 \quad (3.6)$$

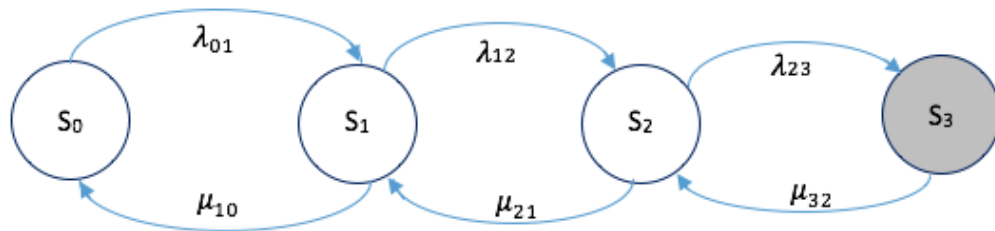
với  $p(t) = e^{-\lambda t}$

#### ***b.2. Trường hợp phần tử có phục hồi***

Xét trường hợp các máy chủ đều có tính chất phục hồi sau khi gặp sự cố, tùy thuộc vào trạng thái làm việc của mỗi phần tử trong hệ, ta giả thiết với bốn trạng thái xảy ra với hệ thống bao gồm:

- Trạng thái  $S_0$ : Hệ thống với ba phần tử hoạt động.
- Trạng thái  $S_1$ : Có hai phần tử hoạt động tốt – một phần tử hỏng.
- Trạng thái  $S_2$ : Có một phần tử hoạt động tốt – hai phần tử hỏng.
- Trạng thái  $S_3$ : Cả ba phần tử đều hỏng – hệ thống ngừng hoạt động.

Áp dụng lý thuyết chuỗi Markov, ta có thể lập được sơ đồ chuyển trạng thái của hệ với các giá trị của xác suất hỏng  $\lambda$  và xác suất phục hồi  $\mu$  được thể hiện như sau:



Hình 3.8: Sơ đồ chuyển trạng thái của hệ với ba phần tử song song

Theo chuỗi Markov, ta xây dựng được hệ phương trình vi phân theo thời gian  $t$  như sau:

$$\begin{cases} P_0'(t) = -\lambda_{01}P_0(t) + \mu_{10}P_1(t) \\ P_1'(t) = -(\mu_{10} + \lambda_{12})P_1(t) + \lambda_{01}P_0(t) + \mu_{21}P_2(t) \\ P_2'(t) = -(\mu_{21} + \lambda_{23})P_2(t) + \lambda_{12}P_1(t) + \mu_{32}P_3(t) \\ P_3'(t) = -\mu_{32}P_3(t) + \lambda_{23}P_2(t) \\ P_0(t) + P_1(t) + P_2(t) + P_3(t) = 1 \end{cases} \quad (3.7)$$

Cùng với giả thiết các máy chủ là đồng nhất do đó xác suất hỏng và xác suất phục hồi của các phần tử trong hệ thống là như nhau, khi đó ta có mối liên hệ giữa các giá trị  $\lambda$  và  $\mu$  như sau:

Do xác suất hỏng của ba phần tử là như nhau:  $\lambda_{01} = 3\lambda$ ,  $\lambda_{12} = 2\lambda$  và  $\lambda_{23} = \lambda$

Tương tự với xác suất phục hồi, ta có:  $\mu_{32} = 3\mu$ ,  $\mu_{21} = 2\mu$  và  $\mu_{10} = \mu$

Khi đó, từ hệ phương trình (3.7) ta có:

$$\begin{cases} P_0'(t) = -3\lambda P_0(t) + \mu P_1(t) \\ P_1'(t) = -(\mu + 2\lambda)P_1(t) + 3\lambda P_0(t) + 2\mu P_2(t) \\ P_2'(t) = -(2\mu + \lambda)P_2(t) + 2\lambda P_1(t) + 3\mu P_3(t) \\ P_3'(t) = -3\mu P_3(t) + \lambda P_2(t) \\ P_0(t) + P_1(t) + P_2(t) + P_3(t) = 1 \end{cases} \quad (3.8)$$

Giả sử trường hợp ban đầu tại  $t = 0$  thì:  $P_0(0)=1$  và  $P_1(0)=P_2(0)=P_3(0)=0$ , tức là tất cả các phần tử đều ở trạng thái tốt, hệ (3.8) được viết lại thành:

$$\begin{cases} -3\lambda P_0 + \mu P_1 = 0 \\ 3\lambda P_0 - (\mu + 2\lambda)P_1 + 2\mu P_2 = 0 \\ 2\lambda P_1 - (2\mu + \lambda)P_2 + 3\mu P_3 = 0 \\ \lambda P_2 - 3\mu P_3 = 0 \\ P_0 + P_1 + P_2 + P_3 = 1 \end{cases} \quad (3.9)$$

Đặt giá trị  $\rho = \frac{\lambda}{\mu}$ , khi đó ta có:  $P_1 = 3\rho P_0$ ,  $P_2 = 3\rho^2 P_0$ ,  $P_3 = \rho^3 P_0$

Phương trình cuối cùng của hệ trở thành:

$$P_0(1 + 3\rho + 3\rho^2 + \rho^3) = 1 \Rightarrow P_0 = \frac{1}{(1+\rho)^3}$$

Đặt  $(1 + \rho)^3 = A$ , khi đó  $P_1=3\rho/A$ ,  $P_2 = 3\rho^2/A$ ,  $P_3 = \rho^3/A$

Do tại trạng thái  $S_3$  các phần tử bị hỏng nên hệ số sẵn sàng của hệ thống chính là tổng của:

$$K_r = P_0 + P_1 + P_2 = \frac{1 + 3\rho + 3\rho^2}{(1 + \rho)^3} \quad (3.10)$$

Trong đó:  $K_r$  là hệ số sẵn sàng của hệ.

$P_i$  là xác suất làm việc ở trạng thái  $i$ .

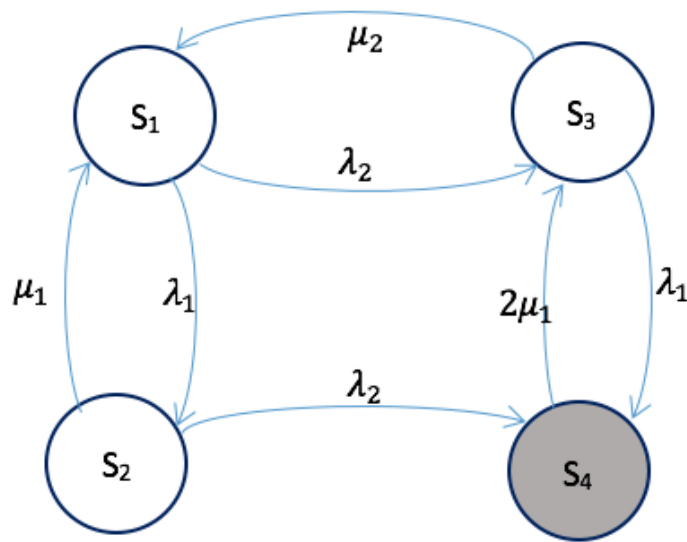
$\rho$  là biểu thức tương quan giữa xác suất hỏng và phục hồi.

### c. Trường hợp phần tử phục hồi có độ ưu tiên

Trong mục a và b, luận án đã đề cập đến bài toán dự phòng cho hệ thống với các phần tử đồng nhất có khả năng phục hồi khi hỏng. Tuy nhiên, trong thực tế không phải lúc nào các thành phần trong hệ thống đều có vai trò như nhau, có các thiết bị hoặc máy chủ khác nhau về cấu hình hoặc vị trí, quan trọng hơn trong việc xử lý các tác vụ của hệ thống, khi đó các phần tử trở thành không đồng nhất.

Trong trường hợp hệ gồm các phần tử không đồng nhất, nghĩa là chúng có thể khác nhau về đặc tính vật lý, hiệu suất hoạt động, tỉ lệ lỗi; Với phần tử có khả năng phục hồi thì tỉ lệ phục hồi cũng sẽ khác nhau. Theo lý thuyết của mô hình chuỗi Markov, với mỗi mô hình chuyển trạng thái khác nhau được xây dựng, sẽ tương ứng với một phương án xảy ra trong quá trình hoạt động của hệ thống. Do vậy, ta có thể xây dựng được các hệ phương trình vi phân khác nhau cho mỗi trường hợp và tính toán được các giá trị độ sẵn sàng, cũng như không sẵn sàng của hệ thống.

Giả thiết với hệ đơn giản nhất gồm hai phần tử hoạt động song song, có tỉ lệ hỏng và tỉ lệ phục hồi khác nhau tương ứng với các giá trị là  $\lambda_1, \lambda_2$  và  $\mu_1, \mu_2$ . Ta có thể xây dựng được sơ đồ chuyển trạng thái như sau:



Hình 3.9: Sơ đồ chuyển trạng thái với hai phần tử phục hồi có ưu tiên khác nhau

Dựa trên sơ đồ thiết kế, hệ thống hoạt động có bốn trạng thái:

- Trạng thái S1: Hệ thống hoạt động; Hai máy chủ PDS và SDS đều cùng hoạt động (trạng thái khởi động);
- Trạng thái S2: Hệ thống hoạt động; Máy chủ PDS bị lỗi, SDS hoạt động bình thường;
- Trạng thái S3: Hệ thống hoạt động; Máy chủ SDS bị lỗi, PDS hoạt động bình thường;
- Trạng thái S4: Hệ thống bị lỗi; Cả hai máy chủ đều bị lỗi (hệ thống ngừng hoạt động).

Tại mỗi trạng thái chuyển dịch trong mô hình Markov, ta có thể thấy khi lỗi xảy ra với mỗi phần tử thì tương ứng với đó là tỉ lệ lỗi  $\lambda$  và khi phần tử phục hồi về trạng thái trước đó, tỉ lệ phục hồi là  $\mu$ .

Theo mô hình chuỗi Markov ta có thể xây dựng được hệ phương trình vi phân của các trạng thái dịch chuyển trong hệ thống như sau:

$$\begin{cases} P_1'(t) = -(\lambda_1 + \lambda_2)P_1 + \mu_1P_2 + \mu_2P_3 \\ P_2'(t) = \lambda_1P_1 - (\lambda_2 + \mu_1)P_2 \\ P_3'(t) = \lambda_2P_1 - (\lambda_1 + \mu_2)P_3 + 2\mu_1P_4 \\ P_4'(t) = \lambda_2P_2 + \lambda_1P_3 - 2\mu_1P_4 \\ P_1 + P_2 + P_3 + P_4 = 1 \end{cases} \quad (3.11)$$

Để giải hệ (3.11), giả sử tại thời điểm ban đầu  $t=0$ : Khi đó  $P_1(0)=1$  và  $P_2(0)=P_3(0)=P_4(0)=0$ . Tức là cả hai phần tử đều ở trạng thái sẵn sàng hoạt động. Khi đó hệ (3.11) trở thành:

$$\begin{cases} -(\lambda_1 + \lambda_2)P_1 + \mu_1 P_2 + \mu_2 P_3 = 0 \\ \lambda_1 P_1 - (\lambda_2 + \mu_1)P_2 = 0 \\ \lambda_2 P_1 - (\lambda_1 + \mu_2)P_3 + 2\mu_1 P_4 = 0 \\ \lambda_2 P_2 + \lambda_1 P_3 - 2\mu_1 P_4 = 0 \\ P_1 + P_2 + P_3 + P_4 = 1 \end{cases} \quad (3.12)$$

Tiến hành đặt giá trị  $\rho_1 = \frac{\lambda_1}{\mu_1}$  và  $\rho_2 = \frac{\lambda_2}{\mu_2}$

Theo mô hình chuyển dịch trạng thái ở Hình 3.9, ta thiết lập được tương quan giữa các trạng thái như sau:

$$P_2 = \rho_1 P_1; P_3 = \rho_2 P_1; P_4 = \frac{1}{2} \rho_1 P_3 = \frac{1}{2} \rho_1 \rho_2 P_1$$

$$\text{Từ đó ta có: } P_1 + P_2 + P_3 + P_4 = P_1(1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2) = 1 \quad (3.13)$$

$$\text{Vậy } P_1 = \frac{1}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2}$$

$$P_2 = \frac{\rho_1}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2}; P_3 = \frac{\rho_2}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2}; P_4 = \frac{\rho_1 \rho_2}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2}$$

Theo trên, hệ số sẵn sàng của hệ được xác định bằng tỉ lệ trung bình thời gian mà hệ thống hoạt động, dựa trên các trạng thái ở đó có ít nhất một phần tử làm việc. Như vậy có ba trạng thái hoạt động trong mô hình đang xét là:  $P_1, P_2, P_3$ .

Hệ số sẵn sàng của hệ:

$$\begin{aligned} A_{s1} &= P_1 + P_2 + P_3 \\ &= \frac{1}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2} + \frac{\rho_1}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2} \\ &+ \frac{\rho_2}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2} = \frac{1 + \rho_1 + \rho_2}{1 + \rho_1 + \rho_2 + \frac{1}{2} \rho_1 \rho_2} \end{aligned} \quad (3.14)$$

Để so sánh khả năng sẵn sàng của hệ các phần tử phục hồi ưu tiên với hệ thống có phục hồi đồng thời, ta tiến hành so sánh với hệ thống gồm 2 phần tử đồng

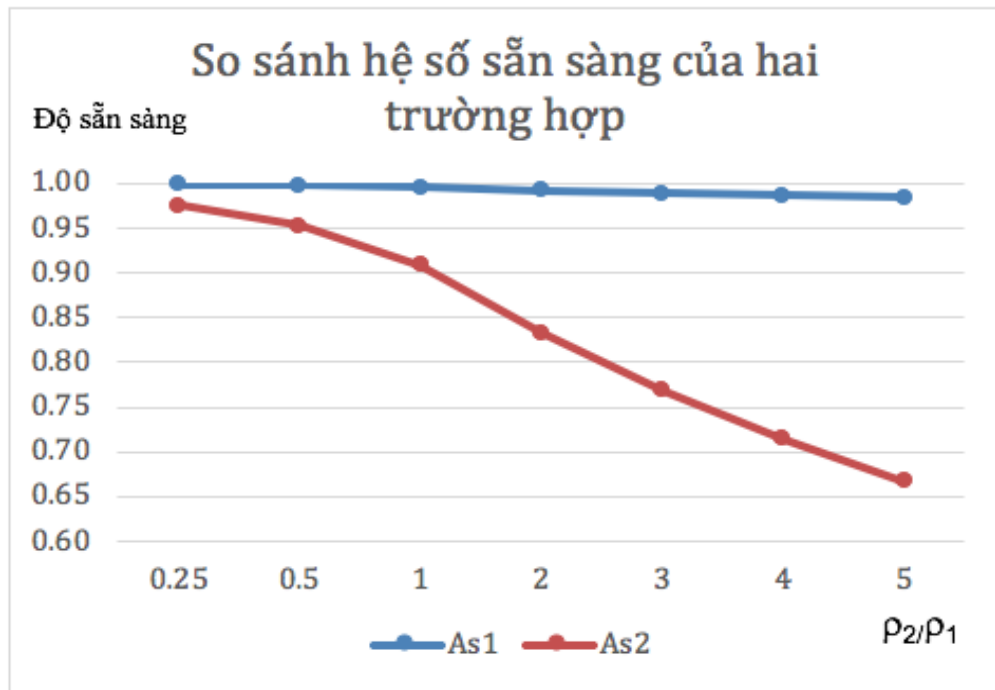
nhất hoạt động song song với hệ số sẵn sàng được xác định trong Công thức (3.5), đặt là  $A_{s2} = \frac{1+2\rho}{(1+\rho)^2}$ , với  $\rho=\lambda/\mu$ .

Thực hiện lập bảng tương quan giữa các giá trị và biểu đồ so sánh hệ số sẵn sàng trong hai trường hợp như sau:

**Bảng 3.1: So sánh hệ số sẵn sàng giữa hai trường hợp phân tử hệ thống.**

$\rho_1$	$\rho_2$	$A_{s1}$	$A_{s2}$
<b>0,1</b>	0,025	0,9988901221	0,9756097561
<b>0,1</b>	0,05	0,9978308026	0,9523809524
<b>0,1</b>	0,1	0,9958506224	0,9090909091
<b>0,1</b>	0,2	0,9923664122	0,8333333333
<b>0,1</b>	0,3	0,9893992933	0,7692307692
<b>0,1</b>	0,4	0,986842105	0,7142857143
<b>0,1</b>	0,5	0,984615385	0,6666666667

Theo đó ta lập được biểu đồ so sánh trong Hình 3.10 dưới đây:



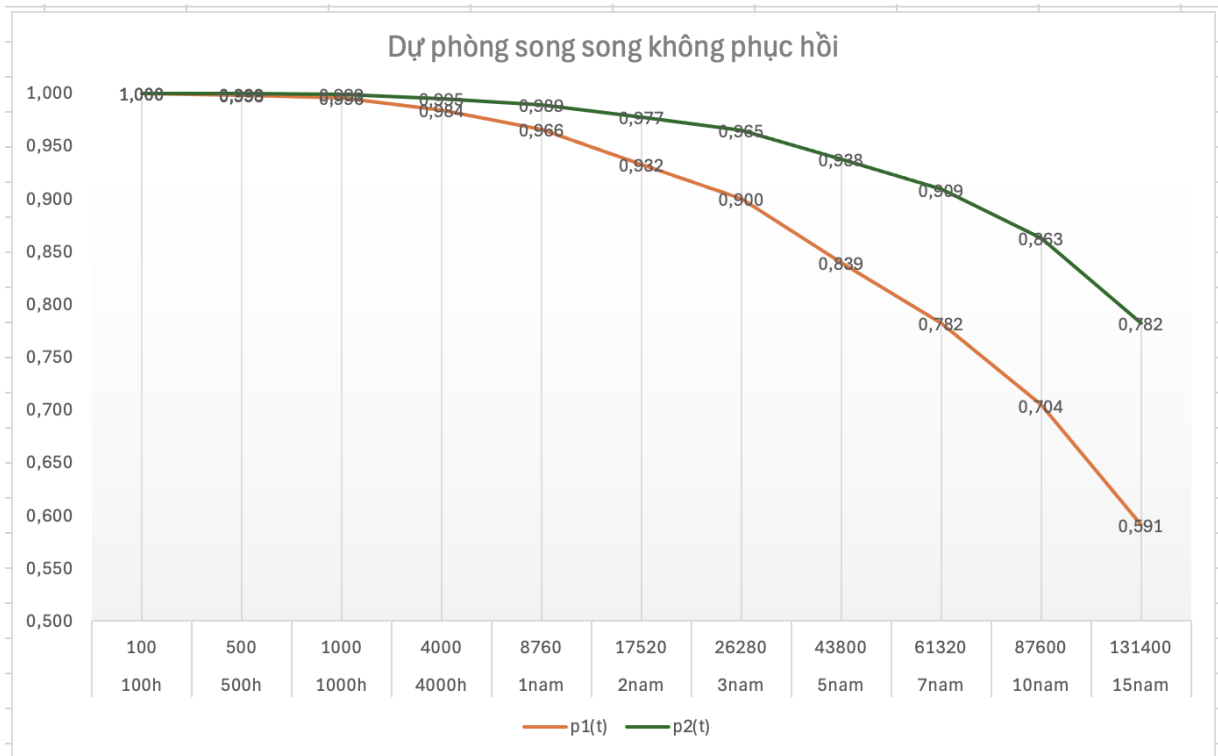
Hình 3.10: Tương quan hệ số sẵn sàng trong hai trường hợp

Dựa trên biểu đồ trực quan, có thể thấy hệ số sẵn sàng của hai hệ thống có sự thay đổi khi giá trị  $\rho_2$  biến thiên. Do ở trên đã cố định giá trị  $\rho_1$ , nên giá trị  $A_{s1}$  và  $A_{s2}$  được tính theo  $\rho_2$ . Khi hệ số  $\rho_2$  biến đổi trong khoảng  $[0,025;0,5]$  thì giá trị của  $A_{s1}$  thay đổi rất ít, khoảng 1,42%, trong khi giá trị của  $A_{s2}$  thay đổi tới 30,8%. Điều này có thể thấy rằng khi chúng ta lựa chọn đúng phần tử để ưu tiên phục hồi (sửa chữa khi bị lỗi), thì khả năng sẵn sàng làm việc lại của toàn bộ hệ thống sẽ cao hơn. Bên cạnh đó, phần tử được lựa chọn phục hồi thường có tiêu chí như: thời gian sống (làm việc) lâu hơn, thời gian sửa chữa nhanh hơn, hiệu suất hoạt động tốt hơn...

#### d. Nhận xét

Trong phương án dự phòng song song với các phần tử không phục hồi, ta đã xác định được hai công thức (3.1) và (3.6) dùng để tính độ tin cậy của hệ thống ở hai trường hợp: sử dụng một phần tử và sử dụng hai phần tử.

Lấy giá trị  $\lambda = 2 \cdot 10^{-7}$  và  $\alpha = 0,8$  theo [22, 53], ta lập được biểu đồ sau:



Hình 3.11: Biểu đồ so sánh các phương án dự phòng với phần tử không phục hồi.

Theo Hình 3.11, phương án sử dụng một phần tử dự phòng (p1) có độ tin cậy giảm nhiều hơn so với phương án sử dụng hai phần tử dự phòng (p2). Độ tin cậy giảm theo từng năm và đặc biệt từ năm thứ nhất đến năm thứ 5:

- Phương án p1: độ tin cậy giảm 13% từ 96% còn 83%; trong 5 năm tiếp theo, độ tin cậy giảm thêm 17%;
- Phương án p2: độ tin cậy trong 5 năm đầu giảm từ 98% xuống 93%; Trong 5 năm tiếp theo, độ tin cậy giảm thêm 13%.

Như vậy, dựa vào biểu đồ và kết quả tính toán, ta có thể xác định được mức độ giảm độ tin cậy của các phương án qua từng khoảng thời gian khác nhau. Đồng thời việc xây dựng công thức tính cho phép áp dụng phương pháp tương tự khi bổ sung thêm các phần tử dự phòng khác vào hệ thống. Ta có thể lượng hóa cụ thể để

đánh giá độ tin cậy hệ thống, từ đó giúp lựa chọn phương án dự phòng tối ưu hơn về thời gian hoặc số lượng phần tử cần thêm.

Trường hợp các phần tử đồng nhất có phục hồi trong hai phương án dự phòng đã khảo sát và tính toán ở mục a.2 và b.2, ta cũng đã xác định được hai công thức dùng xác định hệ số sẵn sàng (3.5) và (3.10).

$$\text{Đặt } K_{r2} = \frac{1+2\rho}{(1+\rho)^2} \text{ và } K_{r3} = \frac{1+3\rho+3\rho^2}{(1+\rho)^3}$$

Ta lập được bảng sau đây:

**Bảng 3.2: So sánh độ tin cậy của hệ thống sau các mốc thời gian**

Thời gian	$\lambda$	$\mu$	$\rho = \frac{\lambda}{\mu}$	$K_{r2}$	$K_{r3}$
<b>Năm 1</b>	$10^{-7}$	$10^{-3}$	0,01	0,999902	0,999995
<b>Năm 3</b>	$10^{-7}$	$10^{-4}$	0,1	0,976709	0,999249
<b>Năm 5</b>	$10^{-7}$	$2 \cdot 10^{-4}$	0,05	0,993413	0,999892
<b>Năm 7</b>	$10^{-7}$	$10^{-5}$	1	0,500000	0,875000

Trong đó:

- $K_{r2}$ : Hệ số sẵn sàng của hệ thống với hai phần tử song song.
- $K_{r3}$ : Hệ số sẵn sàng của hệ thống với ba phần tử song song.

Có thể thấy rằng: công thức tính hệ số sẵn sàng  $K_r$  hai phương án dự phòng đều độc lập với biến thời gian  $t$  và chỉ phụ thuộc vào hệ số  $\rho$ . Trong đó,  $\rho$  là hệ số tương quan giữa xác suất hỏng  $\lambda$  và xác suất phục hồi  $\mu$  của phần tử.

Trong hệ thống với phần tử có khả năng phục hồi cho kết quả độ tin cậy và khả năng sẵn sàng cao cho hệ thống trong khoảng thời gian dài hơn đáng kể so với hệ có phần tử thường. Điều này có thể giải thích thông qua giá trị xác suất hỏng  $\lambda$  và xác suất phục hồi  $\mu$ , với các thiết bị điện tử, tuổi thọ có thể kéo dài tới 5 năm nên giá trị  $MTTF = 1/\lambda$  rất nhỏ, tức là tỉ lệ hỏng, xảy ra lỗi của thiết bị trong những năm đầu tiên của vòng đời hoạt động là rất thấp. Tuy nhiên khi thiết bị gặp sự cố,

thời gian sửa lỗi chính là tốc độ phục hồi, tốc độ phục hồi này thường được tính bằng giờ hoặc ngày, tức là giá trị  $MTTR = 1/\mu$ , do  $\mu \gg \lambda$ , chính vì vậy hệ số sẵn sàng của cả hệ có giá trị rất lớn.

Về nguyên lý làm việc của các thiết bị điện tử trong điều kiện thực tế, sau thời gian 1 đến 2 năm làm việc liên tục, khả năng phục hồi của các thiết bị sẽ giảm xuống không thể duy trì như ở trạng thái mới được, tức là thời gian cần thiết để thiết bị khôi phục hoạt động tăng lên (có thể do lỗi hỏng, do vấn đề bảo trì, thay thế...), nên giá trị  $\mu$  sẽ tăng đáng kể, giả thiết rằng giá trị  $\mu$  sau một năm làm việc sẽ tăng lên  $\mu = 10^{-4}$ . Bên cạnh đó, khả năng xảy ra hỏng của thiết bị sau thời gian vận hành liên tục cũng sẽ có xác suất cao hơn, điều này dẫn đến giá trị hệ số  $\lambda$  tăng lên.

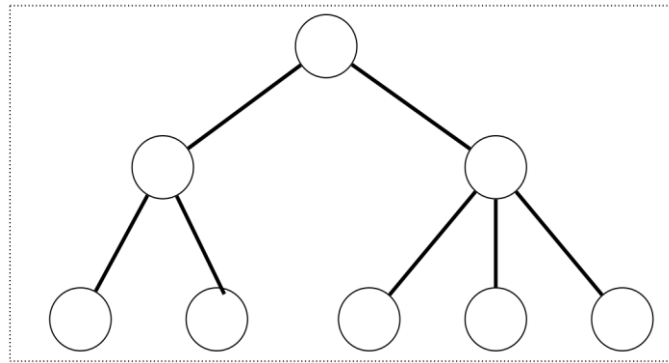
Theo Bảng 3.2, ta có thể thấy rằng hệ số sẵn sàng đã giảm đi sau một năm làm việc. Tuy nhiên việc bổ sung thêm phần tử dự phòng đã cải thiện đáng kể độ tin cậy của hệ thống, việc cần thiết là xác định số lượng phần tử dự phòng cần sử dụng cho phù hợp với yêu cầu với mục đích và chi phí dành cho hệ thống.

### **3.3.2 Đảm bảo độ tin cậy hệ thống sử dụng dự phòng song song**

Với phương pháp đề xuất đảm bảo độ tin cậy trong Mục 3.2.2, luận án tiến hành thử nghiệm với hệ thống máy chủ dịch vụ hoạt động theo mô hình được trình bày trong Hình 3.12, **mô hình này được giả định trên cơ sở lý thuyết với các kết nối và sự ràng buộc giữa các thành phần với nhau về vai trò hoạt động. Tuy nhiên, mô hình hoạt động của hệ thống này hoàn toàn có thể tồn tại trong thực tế ở các trung tâm dịch vụ Internet hay các nhà mạng viễn thông hoặc hệ thống cung cấp dịch vụ dữ liệu.**

Hệ thống có ba cấp độ hoạt động (three-tiers) thường được triển khai trong thực tế để cung cấp dịch vụ đến người dùng cuối. Các cấp độ bao gồm: cấp độ đầu

tiên đóng vai trò như bộ vi xử lý kiểm soát tổng thể; cấp độ thứ hai gồm hai bộ vi xử lý đóng vai trò điều khiển; cấp độ thứ ba gồm bộ các vi xử lý thực hiện tác vụ người dùng cuối.



Hình 3.12: Hệ thống ban đầu không có dự phòng.

Trong mô hình của hệ thống trên, ở mỗi cấp độ đều có các bộ xử lý hoạt động song song, thực hiện nhiệm vụ xử lý dữ liệu và trả kết quả cho các tầng phía trên. Ở tầng cuối cùng trong mô hình, giả thiết có năm bộ vi xử lý hoạt động độc lập và không thể thay thế nhiệm vụ lẫn nhau. Tương tự như vậy, các bộ vi xử lý làm việc ở tầng giữa cũng hoạt động song song, có vai trò khác nhau, do đó nếu một bộ vi xử lý bị lỗi sẽ dẫn đến hệ thống bên dưới bị lỗi theo.

Mục tiêu đề ra đó là từ mô hình của hệ thống ban đầu, áp dụng quy trình đã đề xuất để xây dựng các phương án dự phòng khả thi cho hệ thống, tiếp theo thực hiện xây dựng công thức tính toán độ tin cậy cho từng phương án đã lập và so sánh theo từng mốc thời gian cụ thể. Dựa trên kết quả so sánh để lựa chọn cấu hình dự phòng tối ưu nhất có thể đáp ứng yêu cầu đặt ra và phù hợp với các tiêu chí phục vụ triển khai hệ thống.

Tiến hành thực hiện lần lượt các bước trong quy trình, ta xác định số lượng và vị trí các phần tử dự phòng cần thiết được đặt để có thể thay thế cho các máy

chủ trong mỗi tầng, hoặc thực hiện phương án bổ sung chi nhánh mới để dự phòng cho một nhánh trong hệ thống.

***Bước 1: Xác định yêu cầu của hệ thống***

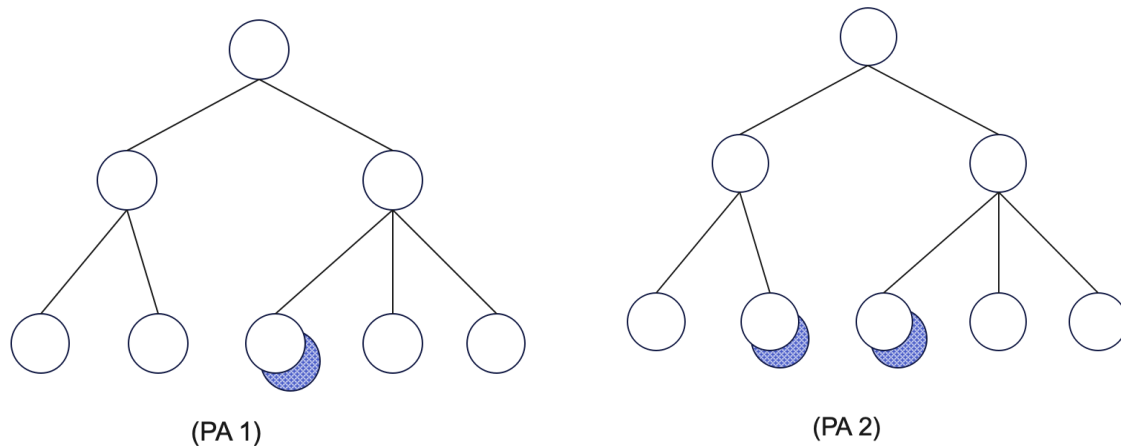
Giả sử với yêu cầu các phần tử trong hệ thống là đồng nhất và không phục hồi, cần xác định phương án thiết lập dự phòng cho các phần tử để hệ thống làm việc ổn định, sau thời gian 5 năm độ tin cậy cần đạt trên 85%.

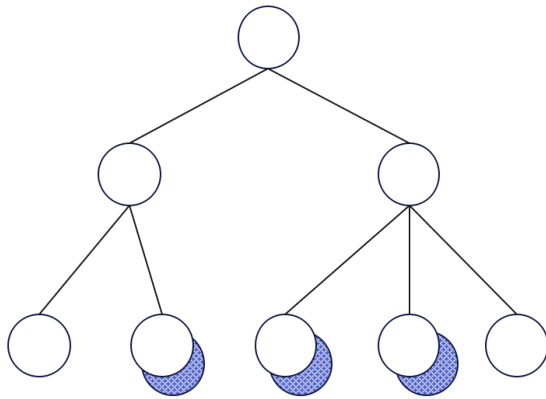
***Bước 2: Kiểm tra mức độ sẵn sàng của hệ thống hiện tại***

Do hệ thống hiện tại có tám phần tử hoạt động đồng thời và không có bất cứ cơ chế dự phòng nào, vì vậy theo lý thuyết độ tin cậy. Ta tính được xác suất độ tin cậy của hệ thống làm việc là:  $P = p^8$ ; Trong đó P là độ tin cậy của hệ thống, còn  $p$  xác suất làm việc của một phần tử bất kỳ trong hệ. Yêu cầu đề ra là thời gian hoạt động trong 5 năm tiếp theo, do đó với cấu hình hiện tại của hệ thống sẽ không thể đảm bảo mức độ tin cậy như vậy.

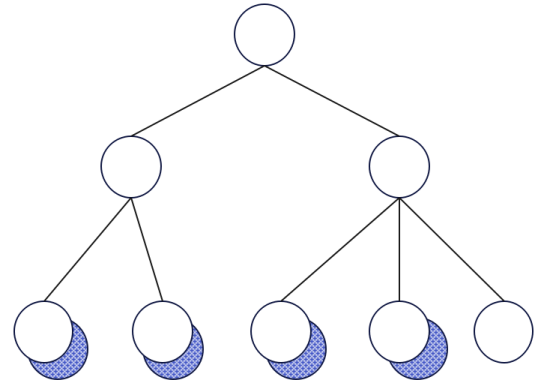
***Bước 3: Xác định các phương án dự phòng có thể cho hệ thống***

Theo cơ chế dự phòng song song, từ cấu trúc ban đầu của hệ thống, ta có các phương án dự phòng như sau:

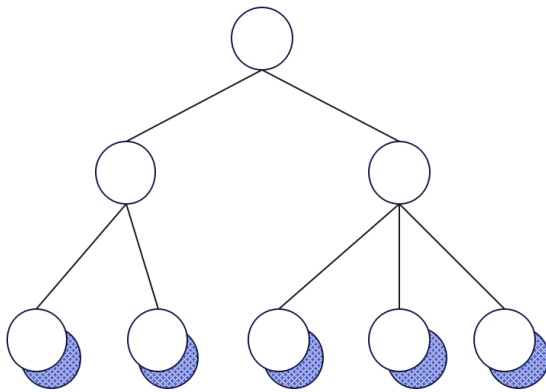




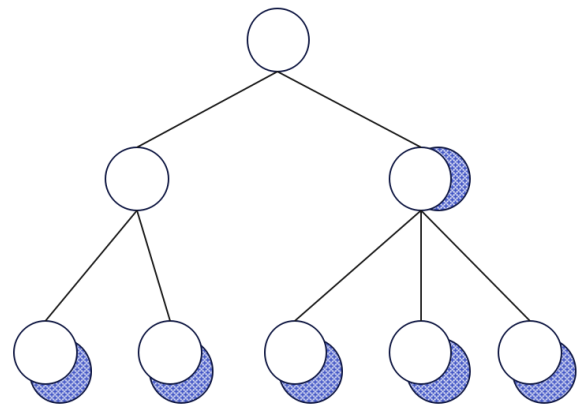
(PA 3)



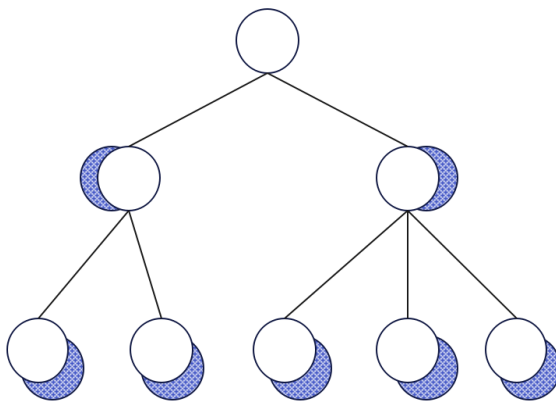
(PA 4)



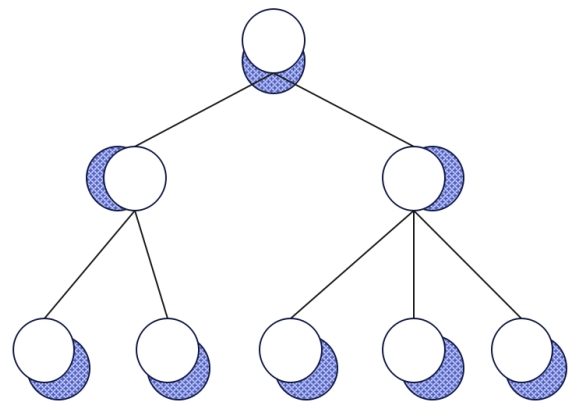
(PA 5)



(PA 6)



(PA 7)



(PA 8)

Hình 3.13: Các phương án dự phòng khả thi của hệ thống.

**Bước 4: Thực hiện tính độ tin cậy cho phương án dự phòng.**

Sử dụng các ký hiệu:

- $p = p(t)$ : Là khả năng hoạt động không có sự thất bại của mỗi bộ xử lý.
- $q = 1-p$  xác suất thất bại của một phần tử.
- $\alpha_i$  khả năng phát hiện chính xác xác suất thất bại tại một mức.
- $P_s$ : Là xác suất hoạt động không hỏng của cả hệ.

Do mỗi cặp có thể coi là một hệ thống gồm hai phần tử độc lập mắc song song là  $X_1, X_2$  với xác suất hoạt động an toàn cùng là  $p$ . Khi đó ta có độ tin cậy của một cặp vi xử lý sẽ là  $P_s$ :

$$\begin{aligned} P(X_1 \vee X_2) &= P(X_1) + Q(X_1).P(X_2) = p + (1-p).p = 2p - p^2 & (3.15) \\ &= 2p(1-p) + p^2 = 2pq + p^2 \end{aligned}$$

$$\text{Với } \alpha_1 \text{ là xác suất thất bại của mỗi cặp, ta có: } P_s = 2\alpha_1 pq + p^2 \quad (3.16)$$

Sau khi biến đổi Công thức (3.16), ta thu được:

$$P_s = 1 - (1-p)[1 - p(2\alpha_1 - 1)] \quad (3.17)$$

Từ các cấu hình hệ thống với bộ vi xử lý dự phòng tại mỗi phương án đã xác lập, biểu thị cho khả năng hoạt động không có sự thất bại hệ thống, ta có thể viết tổng quát hóa như sau:

$$P_s = \prod_{i=0}^N P_i = P^{N-d} P_s^d = P^{N-d} (1 - (1-p)[1 - p(2\alpha_1 - 1)])^d \quad (3.18)$$

Trong đó:

$P_i$  - xác suất thất bại của phần tử thứ  $i$  (bộ xử lý) trong hệ thống

$N$  - số lượng các bộ vi xử lý trong hệ thống

$d$  - số lượng các cặp vi xử lý bản sao trong hệ thống.

$p$  - xác suất hoạt động an toàn của phần tử.

Ứng với các phương án dự phòng đã xây dựng được tại Hình 3.5, ta thiết lập được biểu thức tính xác suất hoạt động không có thất bại của mỗi cấu hình như sau:

$$P = p^8; \text{ (Cấu hình hệ thống ban đầu không dự phòng)}$$

$$P_{(PA1)} = p^7 p_s = p^7 (1 - (1 - p)[1 - p(2\alpha_1 - 1)]);$$

$$P_{(PA2)} = p^6 p_s^2 = p^6 (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^2;$$

$$P_{(PA3)} = p^5 p_s^3 = p^5 (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^3;$$

$$P_{(PA4)} = p^4 p_s^4 = p^4 (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^4;$$

$$P_{(PA5)} = p^3 p_s^5 = p^3 (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^5;$$

$$P_{(PA6)} = p^2 p_s^6 = p^2 (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^6;$$

$$P_{(PA7)} = p p_s^7 = p (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^7;$$

$$P_{(PA8)} = p_s^8 = (1 - (1 - p)[1 - p(2\alpha_1 - 1)])^8;$$

Sau khi xác định của công thức tính độ tin cậy của hệ thống cho các phương án dự phòng tại Bước 3. Ta tiến hành tính giá trị của các  $P_{(PAi)}$  với giả thiết:

Theo Công thức (1.13), xác suất hoạt động của mỗi phần tử được tính bởi:  $P(t) = e^{-\lambda t}$ , trong đó giá trị  $\lambda$  là cường độ hỏng hóc của phần tử được xác định theo các thành phần như sau [71]:

$$\lambda_T = \lambda_{SD} + \lambda_{SU} + \lambda_{DD} + \lambda_{DU} + \lambda_{NE}$$

Trong đó:  $\lambda_T$  là Tổng tỉ lệ hỏng (thất bại) của phần tử;

$\lambda_{SD}$  là Tỷ lệ hỏng được phát hiện an toàn;

$\lambda_{SU}$  là Tỷ lệ hỏng an toàn không được phát hiện;

$\lambda_{DD}$  là Tỷ lệ hỏng được phát hiện nguy hiểm;

$\lambda_{DU}$  là Tỷ lệ hỏng nguy hiểm không được phát hiện;

$\lambda_{NE}$  là Tỷ lệ hỏng không ảnh hưởng.

Một ví dụ về các thông số hỏng của thiết bị máy phát áp lực được ghi nhận trong tài liệu:

Pressure Transmitter – failure rate data (Source: Exida SERH 2015 – 01 sensors – item 1.6.2)		Per 10 <sup>9</sup> hours (FITs)
Fail dangerous detected	$\lambda_{DD}$	260
Fail dangerous undetected	$\lambda_{DU}$	84
Fail safe detected	$\lambda_{SD}$	0
Fail safe undetected	$\lambda_{SU}$	145
No effect failure	$\lambda_{NE}$	135

Hình 3.14: Thông số về tỉ lệ hỏng của thiết bị máy phát.

Tỷ lệ hỏng được ghi nhận sẽ tính trong khoảng thời gian là 10<sup>9</sup> giờ, như vậy với thiết bị máy phát như trên, giá trị cường độ hỏng là:  $\lambda = 6,24 \cdot 10^{-7}$ .

Với các linh kiện điện tử cấu thành lên thiết bị máy tính, giả định lấy giá trị  $\lambda = 7 \cdot 10^{-7}$  trong quá trình tính toán và lấy giá trị xác suất  $\alpha_1 = 0,8$  theo [53].

Ta tính được giá trị trong Bảng 3.3 dưới đây:

**Bảng 3.3: Giá trị độ tin cậy của các phương án sử dụng dự phòng song song**

Cấu hình	1 năm	2 năm	3 năm	4 năm	5 năm	6 năm	7 năm	%
Ban đầu	0,9521	0,9065	0,8631	0,8218	0,7825	0,7450	0,7094	
PA1	0,9556	0,9132	0,8726	0,8338	0,7967	0,7612	0,7272	1,38%
PA2	0,9591	0,9199	0,8821	0,8459	0,8111	0,7777	0,7456	2,79%
PA3	0,9626	0,9266	0,8918	0,8582	0,8258	0,7945	0,7644	4,21%

PA4	0,9662	0,9334	0,9015	0,8707	0,8407	0,8117	0,7836	5,66%
PA5	0,9697	0,9402	0,9114	0,8833	0,8560	0,8293	0,8034	7,14%
PA6	0,9733	0,9471	0,9214	0,8962	0,8715	0,8473	0,8236	8,65%
PA7	0,9768	0,9540	0,9314	0,9092	0,8873	0,8657	0,8444	10,18%
PA8	0,9804	0,9610	0,9416	0,9224	0,9034	0,8844	0,8657	11,74%

### ***Bước 5: Lựa chọn phương án khả thi và ghi nhận kết quả***

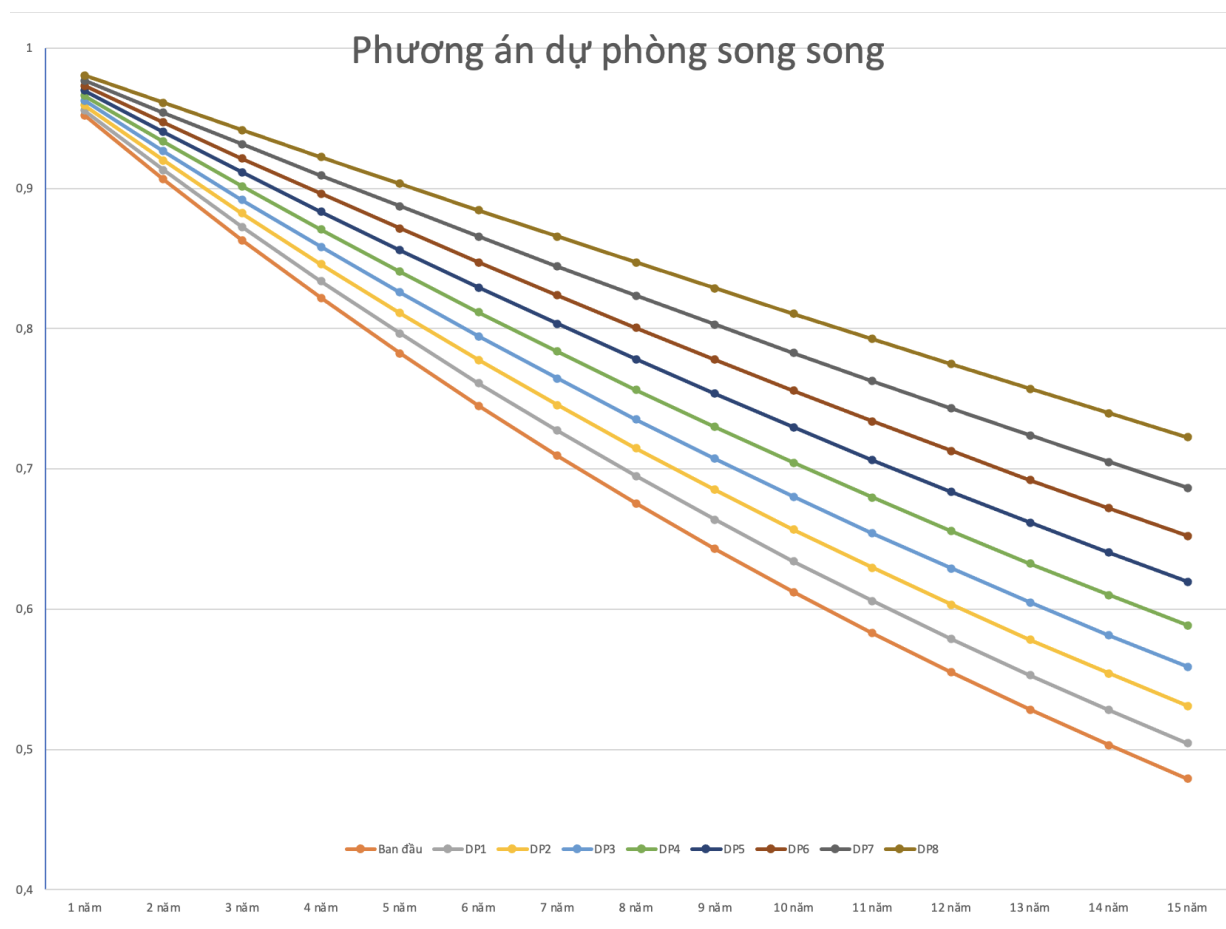
Dựa trên dữ liệu về độ tin cậy tính toán được của các phương án dự phòng theo thời gian từ năm thứ nhất đến năm thứ bảy, ta có thể thấy tỉ lệ phần trăm giữa các phương án dự phòng so với cấu hình ban đầu của hệ thống khi không sử dụng phần tử dự phòng nào.

Với kết quả này, ta có thể lựa chọn phương án dự phòng với độ tin cậy tốt nhất theo tiêu chí đề ra như:

- Đảm bảo độ tin cậy cho hệ thống sau 5 năm sử dụng là trên 90%: chỉ có phương án dự phòng (PA8) đáp ứng được.
- Đảm bảo độ tin cậy cho hệ thống sau 5 năm không nhỏ hơn 85%: ta có thể lựa chọn cấu hình từ (PA7) và (PA8).
- Sau 7 năm hoạt động, mức độ tin cậy của hệ thống cần đạt trên 80%: ta có các cấu hình đáp ứng là (PA5), (PA6), (PA7) và (PA8).

Như vậy tùy thuộc vào tiêu chí về độ tin cậy được nhà thiết kế đặt ra lúc ban đầu, dựa vào quy trình ba bước ở trên có thể xác định và tính toán độ tin cậy cho các phương án dự phòng, từ đó tìm ra được cấu hình phù hợp nhất đáp ứng được điều kiện đặt ra.

Từ Bảng 3.3, ta vẽ được biểu đồ của giá trị độ tin cậy trong các phương án dự phòng khi sử dụng phương án dự phòng song song như Hình 3.15. Có thể thấy rằng: Giá trị độ tin cậy tuân theo quy tắc của hàm mũ  $P(t) = e^{-\lambda t}$ , do vậy các đường của đồ thị có xu hướng giảm dần về 0 khi thời gian kéo ra  $\infty$ . Trong biểu đồ, mốc thời gian khảo sát được tính đến 15 năm cho thấy mức độ tin cậy của hệ thống ở các phương án dự phòng có xu hướng giảm theo từng năm. Trong vòng đời hoạt động, tùy từng yêu cầu của hệ thống, với các phương án mà độ tin cậy giảm xuống dưới 60% thường sẽ phát sinh hỏng và cần thiết phải thay thế trước khi hệ thống không thể đáp ứng yêu cầu vận hành.



Hình 3.15: So sánh độ tin cậy của các phương án dự phòng từ Bảng 3.3.

### 3.4. Nâng cao độ tin cậy sử dụng phương pháp dự phòng tích cực

#### 3.4.1 Tính độ tin cậy hệ thống với dự phòng tích cực

Giả sử tất cả các phân tử hoạt động chính và phân tử dự phòng của hệ trong Hình 3.2 đều đang làm việc trong chế độ có tải. Khi đó, hệ thống sẽ duy trì khả năng hoạt động của mình cho đến khi số lượng các phân tử ở trạng thái hoạt động (trong tổng số  $m+n$  thành phần) không nhỏ hơn  $n$ .

Gọi xác suất không hỏng của cơ cấu chuyển tiếp là  $P_c(t)$ ;  $P(t)$  là xác suất không hỏng của phân tử đứng riêng. Xác suất làm việc không hỏng của phân tử dự phòng đứng trong hệ bằng  $P_c(t)P(t)$ .

Xác suất làm việc không hỏng của hệ thống dự phòng được tính theo [71] xác suất đầy đủ có dạng [89, 90]:

$$P_{mn}(t) = \sum_{i=n}^{m+n} \sum_{k=i-n}^n C_n^{i-k} C_m^k P^{i-k}(t) [1 - P(t)]^{n-i+k} * [P_c(t)P(t)]^k * [1 - P_c(t)P(t)]^{m-k} \quad (3.18)$$

Nếu cơ cấu chuyển tiếp làm việc ở mức độ tin cậy tuyệt đối tức là  $P_c(t)=1$  thì

$$P_{mn}(t) = \sum_{i=n}^{m+n} \sum_{k=i-n}^n C_n^{i-k} C_m^k P^i(t) [1 - P(t)]^{m+n-i} \quad (3.19)$$

vì

$$\sum_{k=i-n}^n C_n^{i-k} C_m^k = C_{m+n}^i \quad (3.20)$$

Nên có thể viết Công thức (3.19) như sau:

$$P_{mn}(t) = \sum_{i=n}^{m+n} C_{m+n}^i P^i(t) [1 - P(t)]^{m+n-i} \quad (3.21)$$

Công thức có thể viết dưới dạng:

$$P_{mn}(t) = 1 - \sum_{i=0}^{n-1} C_{m+n}^i P^i(t) [1 - P(t)]^{m+n-i} \quad (3.22)$$

Trong trường hợp hệ thống có  $n=1$  thì hệ thống dự phòng này trở thành hệ thống dự phòng song song. Với  $n=2$ , áp dụng Công thức (3.22) ta tính được độ tin cậy của hệ thống  $P_{mn}(t)$ :

$$P_{mn}(t) = 1 - \sum_{i=0}^{n-1} C_{m+n}^i P^i(t) [1 - P(t)]^{m+n-i} \quad (3.23)$$

và trong trường hợp số phần tử dự phòng  $m=1$

$$P_{mn}(t) = P^n(t) + n[1 - P(t)]P^n(t) \quad (3.24)$$

Trong các phần tiếp theo, ta sẽ sử dụng các công thức trên để xác định độ tin cậy trong các hệ thống có sử dụng dự phòng tích cực.

### 3.4.2 Bài toán lưu trữ dữ liệu an toàn

Hiện nay, các dịch vụ lưu trữ đám mây như Dropbox, Box, Google Drive, Microsoft 1Drive cung cấp cho người dùng cơ chế lưu trữ linh hoạt, tiện lợi và miễn phí. Dữ liệu có thể được đồng bộ hóa trên nhiều nền tảng bao gồm Web, Mobile và Desktop. Tuy nhiên, điều quan trọng là phải quan tâm đến vấn đề bảo mật dữ liệu vì có những mối đe dọa như mất tài khoản, tuổi thọ của nhà cung cấp dịch vụ và mất quyền truy cập vào tài khoản do hệ thống của nhà cung cấp bị tấn công [17, 58, 77].

Dữ liệu của người dùng luôn là mục tiêu quan tâm của các chính phủ, Google đã nhận hơn 21.000 yêu cầu từ chính phủ Hoa Kỳ để cung cấp thông tin của hơn 33.000 người dùng. Các công ty công nghệ khác như Microsoft cũng ghi nhận hơn 70.000 yêu cầu liên quan đến 122.000 tài khoản người dùng trong hệ thống lưu trữ của họ. Các số liệu này được trích từ phân tích về bảo mật lưu trữ đám mây của Lucas Mearian [58].

Các mối đe dọa mất mát dữ liệu không chỉ là vấn đề của nhà cung cấp dịch vụ mà còn là của người dùng. Tin tặc sử dụng người dùng như con mồi để khai thác thông tin bằng cách giả mạo. Họ tận dụng lỗ hổng của Google Drive – dịch vụ đám mây đáng tin cậy của Google – để thu thập dữ liệu của nạn nhân. Một nhóm tin tặc Trung Đông đã triển khai một vụ lừa đảo lớn qua mạng vào tháng 7 năm 2015 bằng cách sử dụng phương pháp này [77, 88].

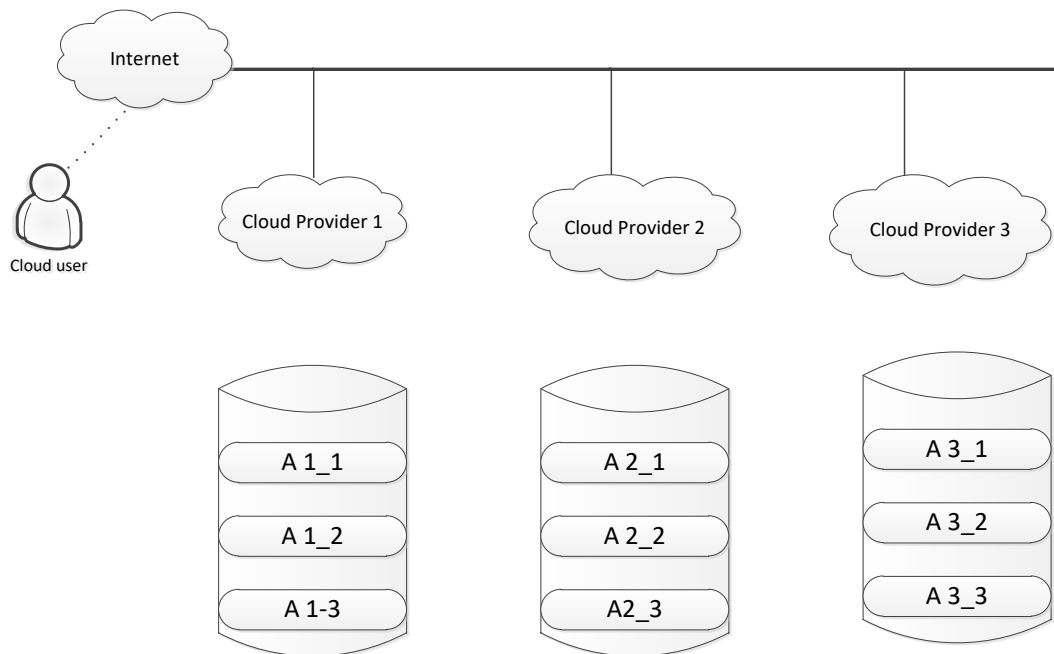
Để nâng cao tính bảo mật cho dữ liệu của người dùng khi sử dụng các hệ thống lưu trữ trên điện toán đám mây, đặc biệt là các dịch vụ được cung cấp miễn phí, tiết kiệm chi phí cho người dùng. Cơ chế lưu trữ dữ liệu dựa trên ý tưởng của mô hình RAID, kết hợp với phương pháp dự phòng tích cực, dữ liệu được thực hiện mã hóa để tránh việc khai thác của tin tặc.

Dựa trên cách thức lưu trữ của **RAID10**, cơ chế lưu trữ dữ liệu RESCS – RAID Enhance Security Cloud Storage được đề xuất [CT1], trên cơ sở kết hợp với phương án dự phòng tích cực, đây là cơ chế lưu trữ dữ liệu sử dụng chính các dịch vụ đám mây miễn phí từ các nhà cung cấp như Google Drive, Dropbox, Box và OneDrive trong việc lưu trữ dữ liệu, đồng thời cũng để làm dự phòng khi xảy ra hỏng. Giải pháp của RESCS kết hợp cơ chế lưu trữ bảo mật với sao lưu **RAID0**, **RAID1**, tính linh hoạt của dịch vụ lưu trữ đám mây, cùng với mã hóa dữ liệu, từ đó cung cấp giải pháp có thể giải quyết hai vấn đề chính khi lưu trữ đám mây:

- Tính toàn vẹn: Dữ liệu được lưu trữ trên một số máy chủ của nhà cung cấp dịch vụ và không hoàn toàn phụ thuộc vào bất kỳ nhà cung cấp nào; do đó, khi một tài khoản không thể truy cập được, những tài khoản khác vẫn có thể tiếp tục hoạt động;
- Tính bảo mật: Sử dụng tính năng phân mảnh dữ liệu để chia dữ liệu thành các khối khác nhau để lưu trữ trên đám mây của các nhà cung cấp khác

nhau, RESCS đảm bảo an toàn cho thông tin trước các cuộc tấn công hoặc truy cập trái phép từ các nhà cung cấp dịch vụ.

Tư tưởng của RESC là: người dùng sử dụng các tài khoản trên các dịch vụ lưu trữ đám mây như Google drive, Dropbox, OneDrive... Để đảm bảo tính toàn vẹn dữ liệu và bảo mật, nên sử dụng ít nhất ba nhà cung cấp dịch vụ khác nhau và số lượng tài khoản trên mỗi dịch vụ tối thiểu là 2 ( $n \geq 2$ ). Như vậy tổng số tài khoản mà mỗi người dùng cần đăng ký là  $3 \cdot n$ . Mô hình hoạt động của hệ thống RESCS như Hình 3.16 dưới đây:



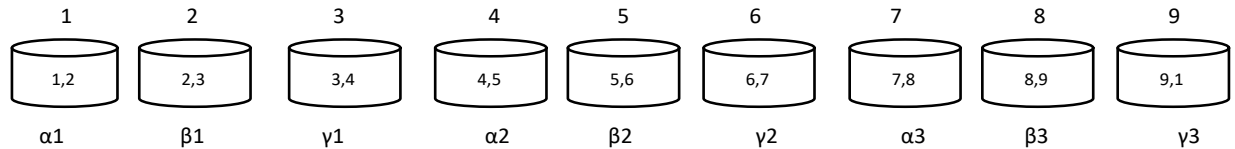
Hình 3.16: Mô hình hoạt động của cơ chế lưu trữ RESCS

Ở đây các ký hiệu  $A_{i_j}$  tương ứng với các tài khoản được tạo trên các nhà cung cấp dịch vụ thứ  $i$ .

Do trong cơ chế dự phòng tích cực này, số lượng phân tử dự phòng  $m$  có thể là bất kỳ tùy vào thiết kế hệ thống. Ý tưởng của RESCS là đặt  $m$  mảnh ghép dữ liệu theo thứ tự lưu trữ trên từng tài khoản như cơ chế hoạt động của RAID. Ta xem xét hai trường hợp:

**Trường hợp m=2: Mỗi tài khoản lưu trữ hai khối dữ liệu**

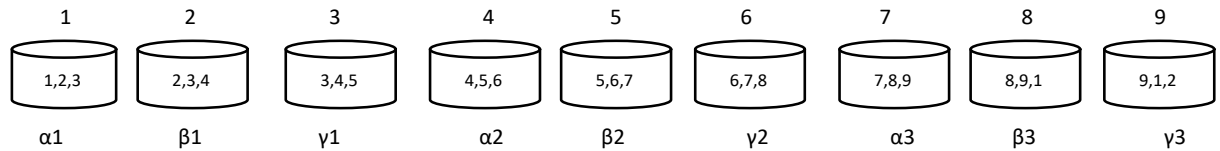
Trong trường hợp này, khi các tài khoản xen kẽ bị mất, dữ liệu có thể được khôi phục từ các tài khoản xung quanh. Nếu nhà cung cấp dừng dịch vụ, dữ liệu sẽ được an toàn nhờ các tài khoản liền kề. Mô hình có thể được biểu diễn như sau:



Hình 3.17: Trường hợp sử dụng hai phần tử dự phòng tích cực

**Trường hợp m=3: Mỗi tài khoản lưu trữ ba khối dữ liệu**

Nếu hai tài khoản liền kề bị mất hoặc không truy cập được thì có thể lấy lại dữ liệu từ các tài khoản lân cận. Nếu một nhà cung cấp ngừng dịch vụ, dữ liệu từ các tài khoản lân cận có thể được sử dụng thay thế. Mô hình hoạt động như sau:



Hình 3.18: Trường hợp sử dụng ba phần tử dự phòng

Coi  $P_{ss}$  là độ tin cậy của hệ thống, ký hiệu:  $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3$  tương ứng là độ tin cậy của từng khối trong các khối trên.

Độ tin cậy của hệ thống được tính theo:

$$P_{ss} = \alpha_1 * \alpha_2 * \alpha_3 * \beta_1 * \beta_2 * \beta_3 * \gamma_1 * \gamma_2 * \gamma_3 \tag{3.25}$$

Giả sử:  $\alpha_1 = \alpha_2 = \alpha_3, \beta_1 = \beta_2 = \beta_3$  và  $\gamma_1 = \gamma_2 = \gamma_3$

Như vậy Công thức (3.25) trở thành:

$$P_{ss} = \alpha^3 * \beta^3 * \gamma^3 \tag{3.26}$$

Áp dụng Công thức (3.26) vào trường hợp m = 2, ta có công thức tính độ tin cậy của hệ thống là:

$$\begin{aligned}
P_{ss1} = & \alpha^3 * \beta^3 * \gamma^3 + (1 - \alpha)^3 * \beta^3 * \gamma^3 + \alpha^3 * (1 - \beta)^3 * \gamma^3 + \alpha^3 \\
& * \beta^3 * (1 - \gamma)^3 + \alpha * \beta^2 * \gamma^2 * (1 - \alpha)^2 * (1 - \beta) \\
& * (1 - \gamma) + \alpha^2 * \beta^2 * \gamma * (1 - \alpha) * (1 - \beta) * (1 - \gamma)^2 \\
& + \alpha^2 * \beta * \gamma^2 * (1 - \alpha) * (1 - \beta)^2 * (1 - \gamma)
\end{aligned} \tag{3.27}$$

Với  $m = 3$ , độ tin cậy của hệ thống trong trường hợp này được tính như sau:

$$\begin{aligned}
P_{ss2} = & \alpha^3 * \beta^3 * \gamma^3 + (1 - \alpha)^3 * \beta^3 * \gamma^3 + \alpha^3 * (1 - \beta)^3 * \gamma^3 + \alpha^3 \\
& * \beta^3 * (1 - \gamma)^3 + (1 - \alpha)^3 * (1 - \beta)^3 * \gamma^3 + \alpha^3 \\
& * (1 - \beta)^3 * (1 - \gamma)^3 + (1 - \alpha)^3 * \beta^3 * (1 - \gamma)^3
\end{aligned} \tag{3.28}$$

Giả sử hai trường hợp này có độ tin cậy cao, nhưng nếu bất kỳ nhà cung cấp dịch vụ lưu trữ nào thay đổi chính sách bảo mật hoặc hệ thống bị tin tặc tấn công thì độ tin cậy của hệ thống sẽ giảm đi xuống do việc kết nối các dịch vụ bị ảnh hưởng.

Theo [53], giả sử lấy giá trị:  $\alpha = 0,9999$  tỉ lệ tài khoản bị lỗi 1/10000,  
 $\beta = 0,9999$  tỉ lệ tài khoản bị lỗi 1/10000,  
 $\gamma = 0,8$  tỉ lệ tài khoản bị lỗi 2000/10000

Khi đó theo áp dụng các giá trị vào Công thức (3.27) và (3.28), ta tính được độ tin cậy ở trường hợp  $m=3$  có sự cải thiện hơn so với  $m=2$  ở mức **1,56%**. Trong trường hợp giữ nguyên giá trị xác suất hỏng của  $\alpha, \beta$ , nhưng giá trị xác suất hỏng của nhà cung cấp dịch vụ còn lại có sự thay đổi: lấy  $\gamma = 0,75$  với tỉ lệ lỗi là 25000/100000. Khi đó, độ tin cậy giữa hai phương án có sự chênh lệch là **3,7%**.

Việc sử dụng cơ chế dự phòng tích cực trong hệ thống RESCS đã giúp cải thiện độ tin cậy và tính sẵn sàng của hệ thống, bên cạnh đó cơ chế lưu trữ phân tán trên các tài khoản khác nhau và phân mảnh dữ liệu kết hợp với mã hóa các khối sẽ tăng đáng kể khả năng bảo mật dữ liệu được lưu trữ trên các dịch vụ đám mây.

### 3.4.3 Đảm bảo độ tin cậy hệ thống với cơ chế dự phòng tích cực

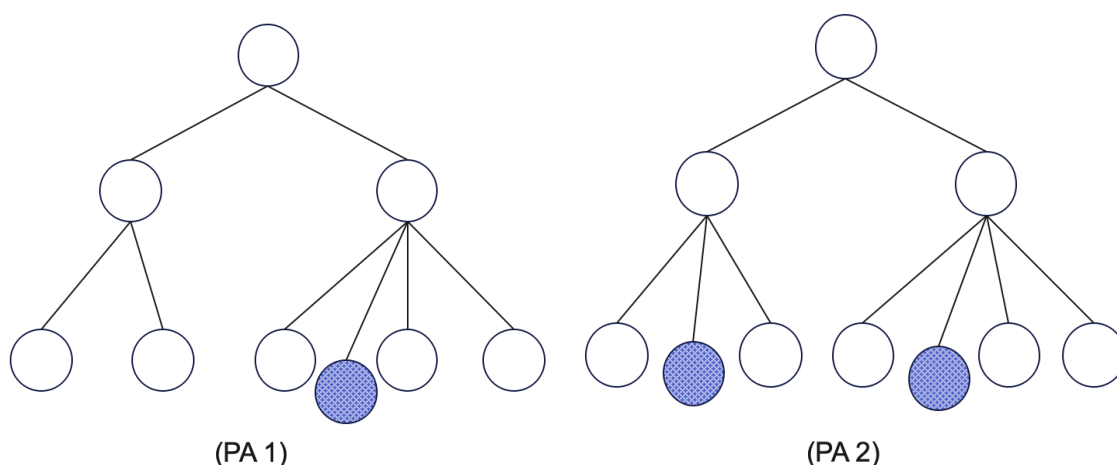
Sử dụng cùng mô hình của hệ thống trong Hình 3.12 với ba cấp độ và tám phần tử tham gia vào quá trình hoạt động của hệ thống. Ta sẽ áp dụng các bước của quy trình đảm bảo độ tin cậy cho hệ thống với việc sử dụng cơ chế dự phòng bảo vệ tích cực.

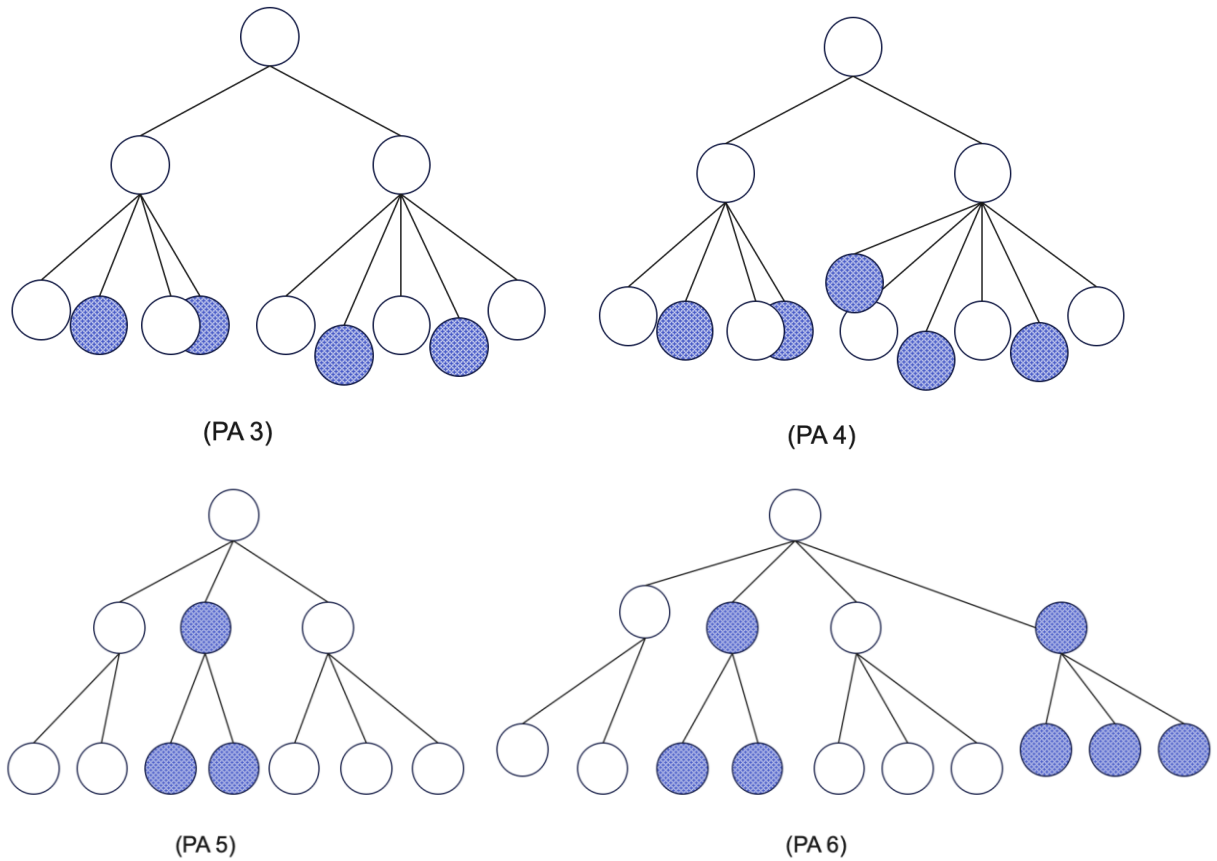
Công thức tổng quát (3.22) sẽ được sử dụng trong các trường hợp cụ thể với hệ thống có số lượng phần tử làm việc chính và dự phòng để tính toán và so sánh độ tin cậy của hệ thống sử dụng cơ chế dự phòng tích cực với các phương pháp truyền thống hoặc chế độ kết hợp.

Áp dụng quy trình đảm bảo độ tin cậy cho hệ thống với phương pháp dự phòng tích cực, các bước thực hiện một và hai đã được tiến hành trong Mục 3.3.1. Tiếp theo, ta cần xác định số lượng và vị trí các phần tử dự phòng cần thiết được đặt để có thể thay thế cho các máy chủ trong mỗi tầng, hoặc thực hiện phương án bổ sung chi nhánh mới để dự phòng cho một nhánh trong hệ thống.

***Bước 3: Xác định các phương án dự phòng có thể cho hệ thống***

Theo cơ chế dự phòng tích cực, ta xác định được một số phương án dự phòng như sau:





Hình 3.19: Các phương án dự phòng tích cực của hệ thống.

**Bước 4: Thực hiện tính độ tin cậy cho phương án dự phòng.**

Trong Hình 3.19, với các cấu hình của mỗi phương án dự phòng, ta sẽ thực hiện việc tính độ tin cậy theo Công thức (3.22).

Sử dụng các ký hiệu:

- $p = p(t)$ : Là khả năng hoạt động không có sự thất bại của mỗi phần tử.
- $q = 1-p$  xác suất thất bại của một phần tử.
- $\alpha_i$  khả năng phát hiện chính xác xác suất thất bại tại một mức.
- $P_s$ : Là xác suất hoạt động không hỏng của cả hệ.

**Phương án một (PA1):** Cấu hình này sử dụng duy nhất một phân tử để dự phòng, như vậy toàn bộ năm phân tử còn lại sẽ hoạt động mà không có dự phòng. Theo nguyên lý của xác suất thì xác suất hỏng của hệ này là  $p^5$ .

Khả năng hoạt động không có sự thất bại là một phần của hệ thống  $P_s$  bằng tổng xác suất hoạt động của cả bốn bộ vi xử lý ( $p^4$ ) và xác suất hoạt động của ba trong số bốn bộ vi xử lý còn lại. Khi đó ta có:

$$P_s = p^4 + 4\alpha_1(1-p)p^3 \quad (3.29)$$

Biểu thức biểu thị cho khả năng hoạt động không có sự thất bại của hệ thống với cấu hình thứ nhất có thể viết như sau:

$$P_{(PA1)} = p^5[p^4 + 4\alpha_1(1-p)p^3] \quad (3.30)$$

**Phương án hai (PA2):** Cấu hình trong phương án này sử dụng hai phân tử cho mỗi chi nhánh con trong sơ đồ của hệ thống. Theo cách tính tương tự trên, ta có thể xác định xác suất hoạt động của hệ trong phương án này theo:

$$P_{(PA2)} = p^3[p^3 + 3\alpha_1(1-p)p^2][p^4 + 4\alpha_1(1-p)p^3] \quad (3.31)$$

**Phương án ba (PA3):** Tại phương án này, hệ thống được bổ sung hai phân tử dự phòng cho mỗi nhánh con trong hệ. Như vậy, ngoài xác suất thất bại cho cấp đầu tiên  $\alpha_1$ , hệ thống sẽ có thêm một xác suất thất bại thứ hai là  $\alpha_2$ , có sự phụ thuộc vào xác suất đầu tiên.

Do đó, công thức tính xác suất của hệ trong phương án này trở thành:

$$P_{(PA3)} = p^3[p^4 + 4\alpha_2(1-p)p^3 + 6\alpha_1(1-p)^2p^2][p^5 + 5\alpha_2(1-p)p^4 + 10\alpha_1(1-p)^2p^3] \quad (3.32)$$

Tương tự như vậy, ta có công thức tính xác suất cho phương án thứ tư:

$$P_{(PA4)} = p^3[p^4 + 4\alpha_2(1-p)p^3 + 6\alpha_1(1-p)^2p^2][p^6 + 6\alpha_3(1-p)p^5 + 15\alpha_2(1-p)^2p^4 + 20\alpha_1(1-p)^3p^3] \quad (3.33)$$

Trong phương án thứ năm và thứ sáu, có sự thay thế toàn bộ một nhánh con trong hệ thống, do vậy việc tính xác suất của khả năng hoạt động của hệ thống sẽ trở nên phức tạp hơn:

$$P_{(PA5)} = p[s^2s_1 + 3\alpha_1(1-s)ss_1] \quad (3.34)$$

$$P_{(PA6)} = p[s^2s_1 + 2\alpha_2(1-s)ss_1^2 + 2\alpha_2(1-s_1)s_1s^2 + 6\alpha_1(1-s_1)(1-s)ss_1] \quad (3.35)$$

Trong đó các giá trị  $s$  được tính theo:

$$s = 1 - \sum_{i=1}^3 C_3^i (1-p)^i p^{3-i}; \quad s_1 = 1 - \sum_{i=1}^4 C_4^i (1-p)^i p^{4-i}$$

**Bước 3: Tính toán và so sánh độ tin cậy của mỗi phương án**

Tương tự như trong Mục 3.3.1, ta cũng giả thiết các giá trị xác suất  $\alpha_1=0,8$ ;  $\alpha_2=1-(1-\alpha_1)^2=0,96$ ;  $\alpha_3=1-(1-\alpha_1)^3=0,992$ ; theo tài liệu [6, 53].

Do thời gian hoạt động của mỗi phân tử được tính theo xác suất là:  $P(t) = e^{-\lambda t}$ , tỉ lệ hỏng hóc của phân tử giả sử lấy:  $\lambda=7.10^{-7}h^{-1}$

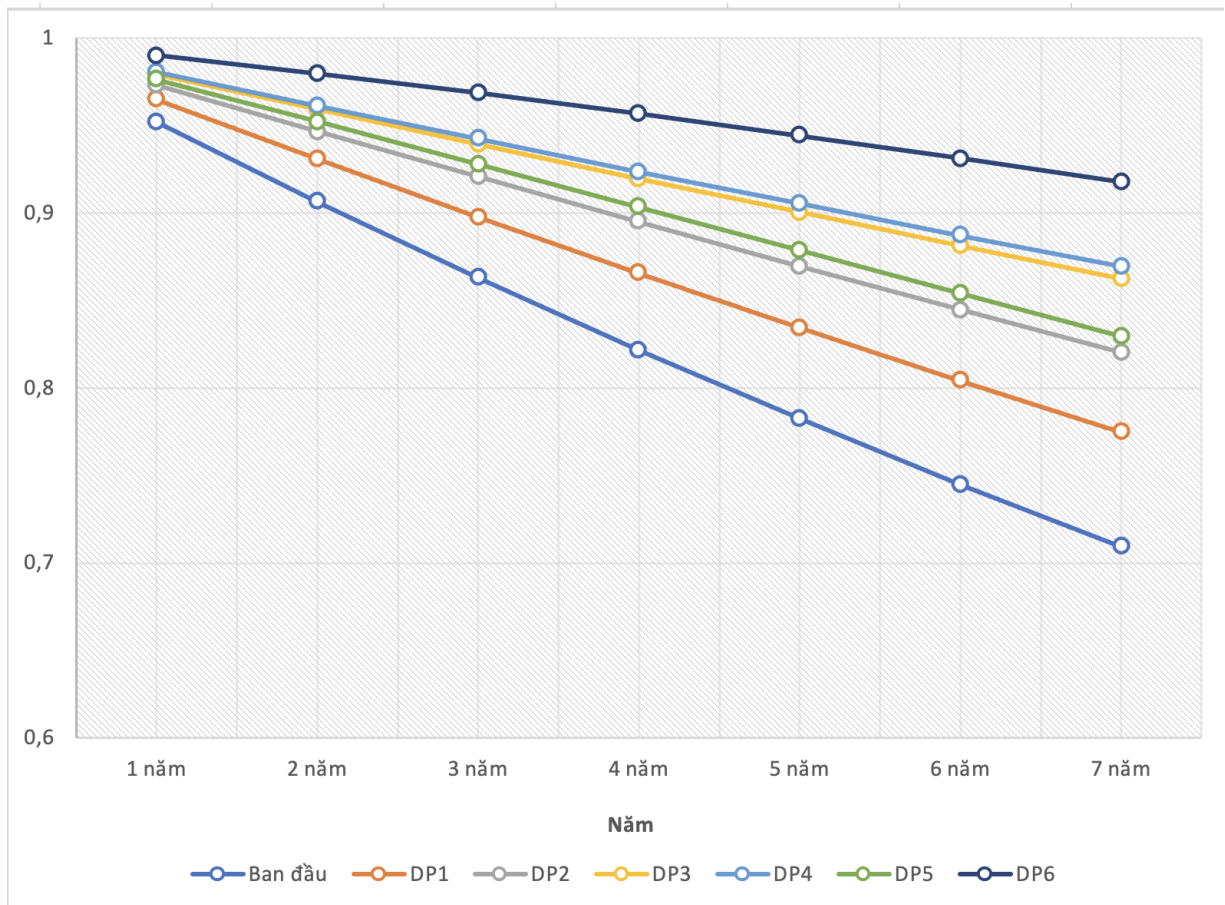
Ta tính được giá trị độ tin cậy của các phương án theo Bảng 3.4 dưới đây:

**Bảng 3.4: Giá trị độ tin cậy của các phương án sử dụng dự phòng tích cực**

Cấu hình	1 năm	2 năm	3 năm	4 năm	5 năm	6 năm	7 năm	%
Ban đầu	0,9521	0,9065	0,8631	0,8218	0,7825	0,7450	0,7094	
(PA1)	0,9649	0,9309	0,8978	0,8656	0,8345	0,8042	0,7749	5,06%
(PA2)	0,9732	0,9467	0,9207	0,8950	0,8697	0,8449	0,8205	8,48%
(PA3)	0,9795	0,9594	0,9395	0,9198	0,9005	0,8814	0,8625	<b>11,46%</b>
(PA4)	0,9805	0,9613	0,9424	0,9237	0,9054	0,8873	0,8696	<b>11,93%</b>
(PA5)	0,9764	0,9524	0,9280	0,9035	0,8788	0,8540	0,8294	9,38%
(PA6)	0,9901	0,9797	0,9686	0,9568	0,9444	0,9313	0,9177	<b>15,71%</b>

**Nhận xét:**

Dựa vào kết quả thu được trong Bảng 3.4, ta lập được biểu đồ tương quan về độ tin cậy sau các năm giữa các cấu hình dự phòng đã tìm được trong Hình 3.9 như sau:



Hình 3.20: So sánh độ tin cậy giữa các phương án dự phòng tích cực

Như vậy với kết quả trên biểu đồ ta có thể thấy sau bảy năm, độ tin cậy của hệ thống với phương án dự phòng thứ sáu (PA6) cho kết quả cao nhất, tiếp theo là các phương án thứ tư (PA4) và thứ ba (PA3).

Dựa vào Bảng 3.4, ta có thể tính được tỉ lệ % chênh lệch giữa các phương án dự phòng, từ đó có thể điều chỉnh bổ sung thêm các phần tử dự phòng cho hệ

thống để đạt được tỉ lệ % độ tin cậy phù hợp theo yêu cầu. Ta có thể lựa chọn các phương án dự án theo các tiêu chí đưa ra ban đầu như:

- Xác định cấu hình dự phòng đảm bảo độ tin cậy trên 90% sau 5 năm đầu tiên: Các cấu hình trong (PA3), (PA4), (PA6) hoàn toàn đáp ứng được.
- Xác định cấu hình dự phòng đảm bảo độ tin cậy trên 90% sau 7 năm làm việc: Chỉ duy nhất cấu hình (PA6) đáp ứng được tiêu chí này.
- **Hãy xác định** cấu hình dự phòng đảm bảo độ tin cậy không nhỏ hơn 85% sau 7 năm hoạt động: Khi đó ta có thể lựa chọn các cấu hình (PA3), (PA4) (PA6) có thể đáp ứng được.

Với cách thức tương tự, ta có thể xác định các cấu hình dự phòng khác bằng việc kết hợp các phương pháp dự phòng khác nhau để đạt được các tiêu chí cao hơn khi thiết kế hệ thống.

### 3.5. Tổng kết chương

Độ tin cậy là một yếu tố cực kỳ quan trọng trong việc quản lý thông tin, đặc biệt trong quản lý hệ thống thông tin hiện đại. Đảm bảo độ tin cậy của hệ thống không chỉ đảm bảo rằng thông tin được bảo vệ một cách an toàn mà còn đảm bảo tính toàn vẹn, sẵn sàng và đáng tin cậy của thông tin.

Nội dung Chương 3 của luận án đã trình bày quy trình đảm bảo độ tin cậy để xác định phương án dự phòng tốt nhất cho hệ thống dựa trên cấu trúc hệ thống và điều kiện ban đầu của bài toán. Tiếp đó, nội dung thực hiện khảo sát và đánh giá độ tin cậy của hệ thống với các trường hợp: hệ thống sử dụng một hoặc hai phần tử dự phòng song song; hệ thống sử dụng phần tử có phục hồi và không có phục hồi; hệ thống có phần tử phục hồi ưu tiên. Qua đó, luận án cũng đã so sánh và phân tích kết quả trong từng trường hợp cụ thể.

Bên cạnh đó, luận án cũng đã thực hiện áp dụng quy trình từng bước đảm bảo độ tin cậy được đề xuất ở đầu chương khi sử dụng với phương pháp dự phòng song song và dự phòng tích cực, từ đó kiểm nghiệm được mức độ tin cậy của hệ thống trong mỗi phương án đưa ra. Các cấu hình dự phòng được xác định sẽ giúp nhà quản trị có căn cứ đánh giá để từ đó xây dựng quy trình trong việc lập kế hoạch lựa chọn phương án tốt nhất, phù hợp với điều kiện thực tế.

Nội dung của chương được tổng hợp và là kết quả của các công trình đã được công bố tại [CT1-CT3], [CT5].

## KẾT LUẬN

Độ tin cậy có vai trò và ý nghĩa hết sức quan trọng trong việc đảm bảo cho hệ thống được hoạt động ổn định, khả năng sẵn sàng trong bất cứ tình huống nào mà còn cho phép người quản trị có thể chủ động các phương án vận hành, duy trì hệ thống. Có nhiều phương pháp để nâng cao độ tin cậy cho hệ thống, trong đó có phương pháp dự phòng. Tuy nhiên việc dự phòng không hiệu quả có thể dẫn đến hao phí về nguồn lực của tổ chức, vì vậy đánh giá độ tin cậy là việc làm hết sức cần thiết để lựa chọn phương án dự phòng phù hợp với mỗi hệ thống.

Nội dung của luận án đã tập trung vào việc đánh giá độ tin cậy của hệ thống thông qua nghiên cứu và so sánh các phương pháp dự phòng, cũng như các biện pháp nâng cao độ tin cậy cho hệ thống máy tính. Các phương pháp dự phòng như dự phòng song song và dự phòng tích cực đã được phân tích và so sánh để xác định hiệu quả của chúng trong việc tăng cường độ tin cậy. Đồng thời, luận án cũng đã triển khai các kỹ thuật và giải pháp mới nhằm nâng cao độ tin cậy của hệ thống, đảm bảo rằng hệ thống máy tính có thể hoạt động ổn định và liên tục, giảm thiểu rủi ro và gián đoạn.

Luận án được trình bày trong năm phần, với ba chương nội dung:

1. Tổng quan về độ tin cậy của hệ thống, các phương pháp được sử dụng để tính độ tin cậy hệ thống, phương pháp đánh giá và dự phòng nâng cao độ tin cậy hệ thống được sử dụng phổ biến hiện nay.

2. Nghiên cứu và đề xuất cải thiện phương pháp tính độ tin cậy giữa hai điểm đầu cuối trong mạng dựa trên phương pháp truyền thống SDP, kết hợp với kỹ thuật tính toán song song để cải thiện hiệu quả tính toán của phương pháp.

3. Nghiên cứu và đề xuất quy trình đảm bảo độ tin cậy của hệ thống dựa trên cơ chế dự phòng, thực hiện đánh giá và so sánh các phương án dự phòng song song, dự phòng tích cực dựa trên cấu trúc của hệ thống.

**Kết quả chính của luận án:**

Thứ nhất, luận án đã đề xuất phương pháp PNRE nhằm cải tiến thuật toán truyền thông SDP để tính độ tin cậy giữa hai thiết bị đầu cuối trong hệ thống mạng. Bằng cách thực hiện song song hóa các hàm tính độ tin cậy của mỗi thành phần con trong đường đi từ điểm nguồn đến đích, phương pháp đã cho kết quả tính toán được cải thiện đáng kể so sánh với hai thuật toán cùng loại là LPC và SACNR.

Thứ hai, đánh giá và đảm bảo độ tin cậy cho hệ thống dựa trên các cơ chế dự phòng. Đề xuất quy trình thực hiện nhằm xác định phương án dự phòng đảm bảo độ tin cậy theo cấu trúc của hệ thống. Kết quả nghiên cứu của luận án là cơ sở để thực hiện việc xác định phương án dự phòng đảm bảo độ tin cậy cho hệ thống hoạt động ổn định.

**Hướng phát triển của luận án:**

Thứ nhất, tiếp tục mở rộng, nghiên cứu phương pháp dự phòng trên các thiết bị IoT với cấu trúc mạng thay đổi. Nghiên cứu và mở rộng phương pháp tính độ tin cậy giữa nhiều phần tử trong hệ thống, thay vì chỉ dừng ở hai thiết bị đầu cuối.

Thứ hai, nghiên cứu và áp dụng các phương pháp đảm bảo độ tin cậy cho hệ thống với các máy chủ trong môi trường điện toán đám mây, nghiên cứu cơ chế tái cấu trúc hệ thống máy ảo để thay thế, dự phòng cho hệ thống ảo hóa.

## DANH MỤC CÔNG TRÌNH TÁC GIẢ ĐÃ CÔNG BỐ

[CT1] Le Quang Minh, Phan Huy Anh, *Nguyen Anh Chuyen*, Le Khanh Duong (2017), “Research on Enhancing Security in Cloud Data Storage”, Proceedings of the International Conference, ICTA 2016, Springer International Publishing, Vol. 538, 510-519. [https://doi.org/10.1007/978-3-319-49073-1\\_55](https://doi.org/10.1007/978-3-319-49073-1_55)

[CT2] *Nguyễn Anh Chuyên* Lê Quang Minh, Lê Khánh Dương, Đinh Thị Thanh Uyên (2017), “Mô hình Markov trong phân tích độ tin cậy của hệ thống máy chủ tên miền DNS Anycast.”, Kỷ yếu hội thảo FAIR lần thứ X, 443-448, ISBN:978-604-913-614-6.

[CT3] *Nguyễn Anh Chuyên*, Lê Quang Minh, Đinh Thị Thanh Uyên, Lê Khánh Dương (2018), “Mô hình Markov trong phân tích độ tin cậy của hệ thống với phần tử phục hồi có độ ưu tiên”, Kỷ yếu hội thảo FAIR lần thứ XI, 262-267, ISBN:978-604-913-749-5.

[CT4]. *Nguyễn Anh Chuyên*, Lê Quang Minh (2023), “An Efficient Method for Evaluating the Two-terminal Reliability with A Parallel Algorithm on the Multi-core Processor Architecture”, International Conference on Information and Communication Technology for Competitive Strategies (ICTCS-2023), 262-267, ISBN: 978-981-99-9485-4.

[CT5]. Lê Quang Minh, *Nguyễn Anh Chuyên* (2024), “Proposing a Process Utilizing Redundancy Methods to Ensure the Reliability of the Server System”, Международный научный журнал «Национальная Ассоциация Ученых», секция "ЕСТЕСТВЕННЫЕ НАУКИ" ISSN Print 2413-5291, ISSN online 2782-2869, pp.34-38.

## CÁC CÔNG BỐ LIÊN QUAN

[CT6]. Lê Quang Minh, *Nguyễn Anh Chuyên*, Lê Khánh Dương, Phan Huy Anh, Trịnh Thị Thu (2016), “Nghiên cứu về các cơ chế RAID và đề xuất giải pháp

lưu trữ dữ liệu an toàn trên dịch vụ đám mây”, Kỹ yếu hội thảo FAIR lần thứ IX, 515-520, ISBN:978-604-913-472-2.

[CT7]. Lê Khánh Dương, Lê Quang Minh, *Nguyễn Anh Chuyên*, Tô Hữu Nguyên (2016), “Đề xuất phương pháp ước lượng độ tin cậy mạng MANET dựa trên kỹ thuật phân cụm và dự phòng mạng”, Kỹ yếu hội thảo FAIR lần thứ IX, 112-117, ISBN:978-604-913-472-2.

[CT8]. Lê Khánh Dương, Nguyễn Văn Tảo, Lê Quang Minh, *Nguyễn Anh Chuyên*, Quách Xuân Trường (2015), “Ảnh hưởng của điều kiện nhiệt độ đối với độ tin cậy của mạng MANET”, Kỹ yếu hội thảo FAIR lần thứ VIII, 30-36, ISBN: 978-604-913-397-8.

[CT9]. Lê Quang Minh, *Nguyễn Anh Chuyên*, Trần Thị Dung (2015), “Nâng cao độ tin cậy cho máy chủ DNS Anycast với giải pháp dự phòng tích cực”, Hội thảo quốc gia một số vấn đề chọn lọc của công nghệ thông tin và truyền thông 2015, tr202-tr206, ISBN 978-604-67-0645-8.

## TÀI LIỆU THAM KHẢO

1. Abouei Ardakan, Mostafa và Ali Zeinal Hamadani (2014), "Reliability optimization of series-parallel systems with mixed redundancy strategy in subsystems", *Reliability Engineering and System Safety*. 130(C), tr. 132-139.
2. Abraham, Jacob A. (1979), "An Improved Algorithm for Network Reliability", *IEEE Transactions on Reliability*. R-28, tr. 58-61.
3. Aslansefat, Koorosh và các cộng sự. (2019), A Markov Process-Based Approach for Reliability Evaluation of the Propulsion System in Multi-rotor Drones, *Technological Innovation for Industry and Service Systems*, Springer International Publishing, Cham, tr. 91-98.
4. Bai, Ya-Nan và các cộng sự. (2019), "Reliability-based topology design for large-scale networks", *ISA Transactions*. 94, tr. 144-150.
5. Balan, A. O. và L. Traldi (2003), "Preprocessing minpaths for sum of disjoint products", *IEEE Transactions on Reliability*. 52(3), tr. 289-295.
6. Ball, Michael O., Charles J. Colbourn và J. Scott Provan (1995), "Handbooks in Operations Research and Management Science", *Handbooks in Operations Research and Management Science*, Elsevier, tr. 673-762.
7. Baroud, Hiba và Kash Barker (2018), "A Bayesian kernel approach to modeling resilience-based network component importance", *Reliability Engineering & System Safety*. 170, tr. 10-19.
8. Billinton R và Wenyuan L (1991), "Hybrid approach for reliability evaluation of composite generation and transmission systems using Monte-Carlo simulation and enumeration technique.", *IEE Proc C*. 138(3), tr. 233-241.
9. Boudali, H. và J. B. Dugan (2005), "A discrete-time Bayesian network reliability modeling and analysis framework", *Reliability Engineering & System Safety*. 87(3), tr. 337-349.
10. Cadini Francesco, Agliardi Gian Luca và Zio Enrico (2017), "A modeling and simulation framework for the reliability/availability assessment of a power transmission grid subject to cascading failures under extreme weather conditions", *Applied Energy*. 185, tr. 267-279.
11. Casali, Alain và Christian Ernst (2013), Extracting Correlated Patterns on Multicore Architectures, Springer Berlin Heidelberg, Berlin, Heidelberg, tr. 118-133.

12. Chatterjee, Subhashis, Venkata Ramana và Gajendra K Vishwakarma (2020), "Analysis of two-terminal network reliability based on efficient data structure", *International Journal of System Assurance Engineering and Management*. 11, tr. 15-20.
13. Chen, Shin-Guang và Yi-Kuei Lin (2009), "On performance evaluation of ERP systems with fuzzy mathematics", *Expert Systems with Applications*. 36(3, Part 2), tr. 6362-6367.
14. Colbourn, Charles J (1987), *The combinatorics of network reliability*, Oxford University Press, Inc.
15. Conrad, David (2012), "Towards improving DNS security, stability, and resiliency", *Internet Society*.
16. Cui, Hongjun và các cộng sự. (2022), "A novel fixed-node unconnected subgraph method for calculating the reliability of binary-state networks", *Reliability Engineering & System Safety*. 226, tr. 108687.
17. D. Zhe, W. Qinghong, S. Naizheng and Z. Yuhan (2017), Study on Data Security Policy Based on Cloud Storage, *iee 3rd international conference on big data security on cloud*, IEEE, tr. 145-149.
18. Daemi, T. và A. Ebrahimi (2012), "Evaluation of Components Reliability Importance Measures of Electric Transmission Systems Using the Bayesian Network", *Electric Power Components and Systems*. 40(12), tr. 1377-1389.
19. Dennis, Norman G (1974), "Reliability analyses of combined voting and standby redundancies", *IEEE Transactions on Reliability*. 23(2), tr. 66-75.
20. Deo, N. và M. Medidi (1992), "Parallel algorithms for terminal-pair reliability", *IEEE Transactions on Reliability*. 41(2), tr. 201-209.
21. El Khadiri, M. và W. C. Yeh (2016), "An efficient alternative to the exact evaluation of the quickest path flow network reliability problem", *Computers & Operations Research*. 76, tr. 22-32.
22. Exida (2015), *Safety Equipment Reliability Handbook - 4th Edition*, Vol. 1, exida.com L.L.C.
23. Feizabadi, Mohammad và Abdolhamid Eshraghniaye Jahromi (2017), "A new model for reliability optimization of series-parallel systems with non-homogeneous components", *Reliability Engineering and System Safety*. 157(C), tr. 101-112.
24. Forghani-elahabad, Majid và Nelson Kagan (2019), "Reliability evaluation of a stochastic-flow network in terms of minimal paths with budget constraint", *IISE Transactions*. 51(5), tr. 547-558.

25. Forghani-elahabad, Majid và Wei-Chang Yeh (2022), "An improved algorithm for reliability evaluation of flow networks", *Reliability Engineering & System Safety*. 221, tr. 108371.
26. Gao, Shan (2023), "Reliability analysis and optimization for a redundant system with dependent failures and variable repair rates", *Mathematics and Computers in Simulation*. 208, tr. 637-659.
27. Gao, Shan, Jinting Wang và Jie Zhang (2023), "Reliability analysis of a redundant series system with common cause failures and delayed vacation", *Reliability Engineering & System Safety*. 239, tr. 109467.
28. Hardy, G., C. Lucet và N. Limnios (2007), "K-Terminal Network Reliability Measures With Binary Decision Diagrams", *IEEE Transactions on Reliability*. 56(3), tr. 506-515.
29. He, Li và Xiaodong Zhang (2016), "Fuzzy reliability analysis using cellular automata for network systems", *Information Sciences*. 348, tr. 322-336.
30. Heidtmann, Klaus (2002), *Statistical Comparison of Two Sum-of-Disjoint-Product Algorithms for Reliability and Safety Evaluation*, Springer Berlin Heidelberg, Berlin, Heidelberg, tr. 70-81.
31. Hosseini, Seyedmohsen và Kash Barker (2016), "Modeling infrastructure resilience using Bayesian networks: A case study of inland waterway ports", *Computers & Industrial Engineering*. 93, tr. 252-266.
32. Hou, K. và các cộng sự. (2016), Composite generation and transmission system reliability assessment using impact increment-based state enumeration method, *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, tr. 1-6.
33. Huang, Cheng-Fu và các cộng sự. (2022), "Network reliability evaluation of manufacturing systems by using a deep learning approach", *Annals of Operations Research*.
34. Huang, Wei, James Loman và Thomas Song (2015), "A reliability model of a warm standby configuration with two identical sets of units", *Reliability Engineering & System Safety*. 133, tr. 237-245.
35. Kai, Hou và các cộng sự. (2016), An impact increments-based state enumeration reliability assessment approach and its application in transmission systems, *2016 IEEE Power and Energy Society General Meeting (PESGM)*, tr. 1-5.

36. Kawahara, Jun và các cộng sự. (2019), "Efficient construction of binary decision diagrams for network reliability with imperfect vertices", *Reliability Engineering & System Safety*. 188, tr. 142-154.
37. Kim, Heungseob và Pansoo Kim (2017), "Reliability models for a nonrepairable system with heterogeneous components having a phase-type time-to-failure distribution", *Reliability Engineering and System Safety*. 159(C), tr. 37-46.
38. Langseth, Helge và Luigi Portinale (2007), "Bayesian networks in reliability", *Reliability Engineering & System Safety*. 92(1), tr. 92-108.
39. Lê, Minh, Max Walter và Josef Weidendorfer (2013), "A Memory-efficient Bounding Algorithm for the Two-terminal Reliability Problem", *Electronic Notes in Theoretical Computer Science*. 291, tr. 15-25.
40. Li, Zhihao và các cộng sự. (2018), Internet anycast: performance, problems, & potential, *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, chủ biên, Association for Computing Machinery, Budapest, Hungary, tr. 59–73.
41. Lin, Y. K., C. H. Huang và S. G. Chen (2021), "Applying Network Reliability in Business Management Activities", *IEEE Access*. 9, tr. 61532-61538.
42. Lin, Yi-Kuei (2002), "Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs", *Reliability Engineering & System Safety*. 75(1), tr. 41-46.
43. Lin, Yi-Kuei (2010), "A novel algorithm to evaluate the performance of stochastic transportation systems", *Expert Systems with Applications*. 37(2), tr. 968-973.
44. Lin, Yi-Kuei và Ping-Chen Chang (2011), "Maintenance reliability estimation for a cloud computing network with nodes failure", *Expert Systems with Applications*. 38(11), tr. 14185-14189.
45. Lin, Yi-Kuei và Ping-Chen Chang (2012), "Reliability evaluation for a manufacturing network with multiple production lines", *Computers & Industrial Engineering*. 63(4), tr. 1209-1219.
46. Lin, Yi-Kuei và Cheng-Fu Huang (2013), "Stochastic Flow Network Reliability with Tolerable Error Rate", *Quality Technology & Quantitative Management*. 10(1), tr. 57-73.
47. Lin, Yi-Kuei và Cheng-Ta Yeh (2012), "Determining the optimal double-component assignment for a stochastic computer network", *Omega*. 40(1), tr. 120-130.

48. Liu, Xiaonan và các cộng sự. (2019), "The Impact-increment State Enumeration Method Based Component Level Resilience Indices of Transmission System", *Energy Procedia*. 158, tr. 4099-4103.
49. Lu, Jiping, Wenyuan Li và Wei Yan (2007), "State enumeration technique combined with a labeling bus set approach for reliability evaluation of substation configuration in power systems", *Electric Power Systems Research*. 77(5), tr. 401-406.
50. M, Lê, M. Walter và J. Weidendorfer (2014), Improving the Kuo-Lu-Yeh Algorithm for Assessing Two-Terminal Reliability, *2014 Tenth European Dependable Computing Conference*, tr. 13-22.
51. Mahadevan, Sankaran, Ruoxue Zhang và Natasha Smith (2001), "Bayesian networks for system reliability reassessment", *Structural Safety*. 23(3), tr. 231-251.
52. Murray, Leslie, Héctor Cancela và Gerardo Rubino (2013), "A splitting algorithm for network reliability estimation", *IIE Transactions*. 45(2), tr. 177-189.
53. NASA (1995), *Military Handbook: Reliability Prediction Of Electronic Equipment*, National Aeronautics And Space Administration.
54. Nguyen, Dang, Bay Vo và Bac Le (2014), "Efficient strategies for parallel mining class association rules", *Expert Systems with Applications*. 41(10), tr. 4716-4729.
55. Nguyen, Thi-Phuong và Yi-Kuei Lin (2021), "Assess reliability of a tourism transport network considering limited-budget and late arrivals", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 236(5), tr. 828-840.
56. Peiravi, Abdossaber và các cộng sự. (2019), "Reliability optimization of series-parallel systems with K-mixed redundancy strategy", *Reliability Engineering & System Safety*. 183, tr. 17-28.
57. Rei, A. M. và M. T. Schilling (2008), "Reliability Assessment of the Brazilian Power System Using Enumeration and Monte Carlo", *IEEE Transactions on Power Systems*. 23(3), tr. 1480-1487.
58. Reilly, Claire (2014), *Hackers hold 7 million Dropbox passwords ransom*, CNET, truy cập ngày, tại trang web <http://www.cnet.com/news/hackers-hold-7-million-dropbox-passwords-ransom>.
59. Rocco S, Claudio M. và José Alí Moreno (2002), "Network reliability assessment using a cellular automata approach", *Reliability Engineering & System Safety*. 78(3), tr. 289-295.

60. Ryabinin, I. A. (2003), "Logical-Probabilistic Calculus: A Tool for Studying the Reliability and Safety of Structurally Complex Systems", *Automation and Remote Control*. 64(7), tr. 1177-1185.
61. Sadeghi, Meisam và các cộng sự. (2020), "Reliability optimization for non-repairable series-parallel systems with a choice of redundancy strategies and heterogeneous components: Erlang time-to-failure distribution", *Proceedings of the Institution of Mechanical Engineers Part O Journal of Risk and Reliability*. 235.
62. Saridou, B., S. Shiaeles và B. Papadopoulos (2019), DDoS Attack Mitigation through Root-DNS Server: A Case Study, *2019 IEEE World Congress on Services (SERVICES)*, tr. 60-65.
63. Sebastio, Stefano và các cộng sự. (2014), "Fast computation of bounds for two-terminal network reliability", *European Journal of Operational Research*. 238(3), tr. 810-823.
64. Shooman, Martin L. (2002), *Reliability Of Computer Systems And Networks: Fault Tolerance, Analysis, and Design*, John Wiley & Sons, Inc.
65. Sihombing, Fritz và Marco Torbol (2018), "Parallel fault tree analysis for accurate reliability of complex systems", *Structural Safety*. 72, tr. 41-53.
66. Soh, S. và S. Rai (1993), "Experimental results on preprocessing of path/cut terms in sim of disjoint products technique", *IEEE Transactions on Reliability*. 42(1), tr. 24-33.
67. Sommese, Raffaele và các cộng sự. (2022), *Investigating the impact of DDoS attacks on DNS infrastructure*, 51-64.
68. Sun, Hong, Yiyang Zhang và Peng Zhao (2022), "Allocating hot standbys to randomly weighted k-out-of-n: G systems", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 236(1), tr. 37-54.
69. Sung, H. và các cộng sự. (2007), Dynamic Clustering Model for High Service Availability, *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)*, tr. 311-317.
70. Sy-Yen, Kuo, Lu Shyue-Kung và Yeh Fu-Min (1999), "Determining terminal-pair reliability based on edge expansion diagrams using OBDD", *IEEE Transactions on Reliability*. 48(3), tr. 234-246.
71. Tacchini, Marco (2023), *Functional Safety of Machinery: How to Apply ISO 13849-1 and IEC 62061*, Wiley; 1st edition, 352.

72. Tsuchiya, T., Kajikawa, T., & Kikuno, T. (2000), Parallelizing SDP (Sum of Disjoint Products) Algorithms for Fast Reliability Analysis, *IEICE transactions on information and systems*, tr. 1183-1186.
73. Vasar, Cristian và các cộng sự. (2009), *Markov Models for Wireless Sensor Network Reliability*, 323-328.
74. Vo, B. và các cộng sự. (2020), "A Multi-Core Approach to Efficiently Mining High-Utility Itemsets in Dynamic Profit Databases", *IEEE Access*. 8, tr. 85890-85899.
75. Volkanovski, Andrija, Marko Čepin và Borut Mavko (2009), "Application of the fault tree analysis for assessment of power system reliability", *Reliability Engineering & System Safety*. 94(6), tr. 1116-1127.
76. Vries, Wouter de (2019), Improving Anycast with Measurements, chủ biên.
77. Whittaker, Zack (2015), *Attackers can access Dropbox, Google Drive, OneDrive files without a user's password*, truy cập ngày Jul 25th-2024, tại trang web <http://www.zdnet.com/article/dropbox-google-drive-onedrive-files-man-cloud-attack>.
78. Wu, Hui, Yan-Fu Li và Christophe BÃ©renguer (2020), "Optimal inspection and maintenance for a repairable k-out-of-n: G warm standby system", *Reliability Engineering and System Safety*. 193(C).
79. Xing, L. (2008), "An Efficient Binary-Decision-Diagram-Based Approach for Network Reliability and Sensitivity Analysis", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*. 38(1), tr. 105-115.
80. Yang, Dong-Yuh và Chia-Huang Wu (2021), "Evaluation of the availability and reliability of a standby repairable system incorporating imperfect switchovers and working breakdowns", *Reliability Engineering & System Safety*. 207, tr. 107366.
81. Yeh, Cheng-Ta và các cộng sự. (2022), "Rail transport network reliability with train arrival delay: A reference indicator for a travel agency in tour planning", *Expert Systems with Applications*. 189, tr. 116107.
82. Yeh, Wei-Chang (2007), "An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths", *Reliability Engineering & System Safety*. 92(2), tr. 260-268.

83. Yeh, Wei-Chang (2008), "An improved algorithm for searching all minimal cuts in modified networks", *Reliability Engineering & System Safety*. 93(7), tr. 1018-1024.
84. Yeh, Wei-Chang (2015), "An Improved Sum-of-Disjoint-Products Technique for Symbolic Multi-State Flow Network Reliability", *IEEE Transactions on Reliability*. 64(4), tr. 1185-1193.
85. Younes, Ahmed và Moheb R. Girgis (2005), "A tool for computing computer network reliability", *International Journal of Computer Mathematics*. 82(12), tr. 1455-1465.
86. Yu, Huan và các cộng sự. (2017), "Reliability evaluation of non-repairable phased-mission common bus systems with common cause failures", *Computers & Industrial Engineering*. 111, tr. 445-457.
87. Zhai, Qingqing và các cộng sự. (2013), "Binary decision diagram-based reliability evaluation of k-out-of-(n + k) warm standby systems subject to fault-level coverage", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 227(5), tr. 540-548.
88. Dung, Xuân (2015), *Phát hiện cuộc tấn công mạng giả mạo Google Drive cực tinh vi*, truy cập ngày, tại trang web <https://kaspersky.proguide.vn/bao-mat-cong-nghe/phat-hien-cuoc-tan-cong-mang-gia-mao-google-drive-cuc-tinh-vi/>.
89. Khôi, Phan Văn (2001), *Cơ sở đánh giá độ tin cậy*, NXB Khoa học kỹ thuật.
90. Minh, Le Quang (2007), "Анализ методов обеспечения отказоустойчивости и живучести вычислительных систем", *Естественные науки и технологии*. 5.