**BÁO CÁO PROJECT**

1. **Giới thiệu**

**Mục đích của ứng dụng SNMP**

SNMP (Simple Network Management Protocol) là một giao thức phổ biến được sử dụng để quản lý và giám sát các thiết bị mạng như bộ định tuyến, switch, máy chủ và các thiết bị IoT. Mục tiêu của dự án là phát triển một ứng dụng SNMP trên nền tảng Java sử dụng thư viện SNMP4j để thực hiện các chức năng quản lý mạng thông qua giao thức SNMP.

Ứng dụng này tập trung vào việc:

1. Cung cấp chức năng SNMP Client và Server.
2. Hỗ trợ các phiên bản SNMPv1, SNMPv2c và SNMPv3 nhằm đảm bảo khả năng tương thích với các thiết bị mạng đa dạng.
3. Đánh giá hiệu năng của ứng dụng thông qua các phép đo benchmark, giúp tối ưu hóa trong triển khai thực tế.

**Tóm tắt yêu cầu và phạm vi project**

Project yêu cầu:

* Xây dựng một ứng dụng Java dựa trên SNMP4j.
* Tích hợp hỗ trợ cho ba phiên bản SNMP chính:
  + **SNMPv1**: Giao thức cơ bản, dễ triển khai nhưng thiếu tính bảo mật.
  + **SNMPv2c**: Cải tiến với hiệu suất tốt hơn nhưng vẫn thiếu bảo mật mạnh mẽ.
  + **SNMPv3**: Cung cấp bảo mật nâng cao với cơ chế xác thực và mã hóa.
* Đánh giá hiệu năng của các chức năng thông qua benchmark, tập trung vào thời gian phản hồi và khả năng xử lý tải.

**Lý do sử dụng SNMP4j**

SNMP4j là một thư viện mã nguồn mở mạnh mẽ, được thiết kế để hỗ trợ các phiên bản SNMP trong môi trường Java. Các lý do chính để lựa chọn SNMP4j bao gồm:

* **Tính tương thích cao**: Hỗ trợ đầy đủ các phiên bản SNMP (v1, v2c, v3).
* **Khả năng bảo mật tốt**: Cung cấp các tính năng bảo mật của SNMPv3 như mã hóa và xác thực.
* **Hiệu suất**: Tối ưu hóa cho các ứng dụng yêu cầu hiệu năng cao.
* **Dễ sử dụng**: Cung cấp các API dễ dàng tích hợp vào các ứng dụng Java.

1. **Tổng quan về SNMP**

**Định nghĩa và vai trò của SNMP**

SNMP (Simple Network Management Protocol) là một giao thức tiêu chuẩn để quản lý và giám sát các thiết bị trong mạng, như bộ định tuyến, switch, máy chủ, và các thiết bị IoT. Được phát triển bởi Internet Engineering Task Force (IETF), SNMP giúp thu thập và tổ chức thông tin từ các thiết bị mạng, đồng thời cho phép cấu hình và điều khiển từ xa.

**Vai trò chính của SNMP**:

1. **Giám sát hiệu năng mạng**: Thu thập dữ liệu hiệu năng như băng thông, trạng thái thiết bị, và thời gian hoạt động.
2. **Phát hiện và xử lý sự cố**: Gửi cảnh báo khi xảy ra lỗi hoặc các vấn đề liên quan đến thiết bị mạng.
3. **Quản lý cấu hình thiết bị**: Điều chỉnh và cập nhật các tham số cấu hình từ xa.
4. **Quản lý tài nguyên mạng**: Theo dõi tài nguyên mạng và tối ưu hóa sử dụng tài nguyên.

#### ****Sự khác biệt giữa các phiên bản SNMP****

SNMP có ba phiên bản chính (v1, v2c, v3), mỗi phiên bản có những đặc điểm riêng:

|  |  |  |  |
| --- | --- | --- | --- |
| Đặc điểm | SNMPv1 | SNMPv2c | SNMPv3 |
| Phát hành | 1988 | 1993 | 2002 |
| Tính năng chính | Đơn giản, dễ triển khai, ít bảo mật | Cải tiến hiệu năng, hỗ trợ thông điệp lớn hơn | Tăng cường bảo mật với mã hóa và xác thực |
| Bảo mật | Không bảo mật | Không bảo mật | Hỗ trợ xác thực (Authentication) và mã hóa (Encryption) |
| Cấu trúc dữ liệu | Cơ bản | Hỗ trợ dữ liệu Bulk Transfer | Tương tự SNMPv2c nhưng bổ sung bảo mật |
| Ứng dụng | Phù hợp với mạng nhỏ, thiết bị cũ | Phổ biến hơn với mạng trung bình | Dành cho mạng lớn, yêu cầu bảo mật cao |
| Tính tương thích | Dễ dàng tương thích với thiết bị cũ | Tương thích ngược với SNMPv1 | Tương thích ngược với SNMPv2c |

**Chi tiết các phiên bản**:

1. **SNMPv1**:
   * Là phiên bản đầu tiên của giao thức, dễ triển khai nhưng hạn chế về khả năng bảo mật.
   * Không có cơ chế xác thực hoặc mã hóa, khiến dữ liệu dễ bị xâm phạm trong môi trường mạng không an toàn.
   * Thích hợp cho các mạng nhỏ hoặc thiết bị cũ.
2. **SNMPv2c**:
   * Cải tiến từ SNMPv1, hỗ trợ truyền tải dữ liệu lớn hơn với hiệu năng tốt hơn.
   * Giới thiệu tính năng Bulk Transfer, giúp giảm số lượng thông điệp cần thiết trong quá trình quản lý.
   * Tuy nhiên, bảo mật vẫn là vấn đề vì không có mã hóa hoặc xác thực.
3. **SNMPv3**:
   * Phiên bản tiên tiến nhất, tập trung vào bảo mật với cơ chế xác thực và mã hóa dữ liệu.
   * Hỗ trợ hai cơ chế bảo mật chính:
     + **Authentication**: Đảm bảo danh tính người gửi.
     + **Encryption**: Bảo vệ nội dung thông điệp khỏi bị nghe lén.
   * Thích hợp cho các hệ thống lớn với yêu cầu bảo mật cao.

#### ****Ưu điểm của việc hỗ trợ đa phiên bản (v1, v2c, v3)****

Ứng dụng hỗ trợ đa phiên bản SNMP mang lại nhiều lợi ích quan trọng:

1. **Khả năng tương thích cao**:
   * Các thiết bị mạng khác nhau có thể hỗ trợ các phiên bản SNMP khác nhau. Việc hỗ trợ đa phiên bản đảm bảo ứng dụng tương thích với cả thiết bị cũ và mới.
2. **Đáp ứng yêu cầu đa dạng của người dùng**:
   * Trong khi một số mạng nhỏ chỉ cần SNMPv1 để quản lý thiết bị cơ bản, các mạng lớn và phức tạp yêu cầu bảo mật nâng cao của SNMPv3.
3. **Tăng cường khả năng quản lý mạng**:
   * Việc hỗ trợ đa phiên bản cho phép ứng dụng linh hoạt hơn trong việc quản lý các thiết bị từ đơn giản đến phức tạp.
4. **Bảo mật và hiệu năng**:
   * Người dùng có thể lựa chọn phiên bản phù hợp tùy theo nhu cầu: SNMPv3 cho bảo mật, SNMPv2c cho hiệu năng cao, hoặc SNMPv1 cho tương thích ngược.
5. **Mở rộng ứng dụng**:
   * Hỗ trợ đa phiên bản giúp ứng dụng trở nên đa năng hơn, dễ dàng mở rộng để đáp ứng nhu cầu của các doanh nghiệp lớn hoặc nhà cung cấp dịch vụ.

### ****3. Chi tiết thiết kế và triển khai****

#### ****Cấu trúc ứng dụng****

**Vai trò của từng file Java**:

1. **SNMPClientBenchmark.java**:
   * Mục đích: Đo lường hiệu năng của SNMP Client khi gửi các yêu cầu đến Server.
   * Chức năng:
     + Tạo các yêu cầu SNMP GET, WALK.
     + Ghi nhận thời gian phản hồi và số lượng yêu cầu xử lý.
2. **SNMPServer.java**:
   * Mục đích: Cung cấp chức năng của một SNMP Server cơ bản (v1 và v2c).
   * Chức năng:
     + Xử lý các yêu cầu SNMP từ client.
     + Gửi phản hồi dữ liệu dựa trên các OID được yêu cầu.
3. **SNMPServerV3.java**:
   * Mục đích: Tương tự như SNMPServer nhưng hỗ trợ SNMPv3.
   * Chức năng:
     + Xử lý các yêu cầu SNMP với cơ chế xác thực và mã hóa.
     + Quản lý thông tin người dùng và bảo mật.
4. **TestSnmpGetV1.java, TestSnmpGetV2c.java, TestSnmpGetV3.java**:
   * Mục đích: Kiểm tra chức năng SNMP Client khi thực hiện các yêu cầu GET với từng phiên bản SNMP.
   * Chức năng:
     + Tạo các yêu cầu GET đến Server.
     + Đánh giá tính chính xác của phản hồi.
5. **TestSnmpWalkV3.java**:
   * Mục đích: Kiểm tra chức năng WALK của SNMPv3.
   * Chức năng:
     + Lấy dữ liệu từ nhiều OID liên tiếp.
     + Đảm bảo cơ chế bảo mật hoạt động đúng.
6. **Constants.java**:
   * Mục đích: Chứa các giá trị cố định dùng chung trong ứng dụng.
   * Chức năng:
     + Quản lý OID, cổng, địa chỉ IP, và các thông số cấu hình khác.
7. **SnmpV3Util.java**:
   * Mục đích: Cung cấp các tiện ích hỗ trợ cho SNMPv3.
   * Chức năng:
     + Tạo và quản lý thông tin bảo mật cho SNMPv3.
     + Hỗ trợ mã hóa và xác thực.
8. **CompareDifferentVersion.java**:
   * Mục đích: So sánh hiệu năng và tính năng của các phiên bản SNMP.
   * Chức năng:
     + Đánh giá thời gian phản hồi, số lượng yêu cầu xử lý.
     + Đưa ra phân tích giữa SNMPv1, v2c và v3

#### ****Mối quan hệ giữa các thành phần trong hệ thống****

* **SNMP Server (SNMPServer.java, SNMPServerV3.java)**:
  + Nhận và xử lý yêu cầu từ SNMP Client.
  + Phản hồi thông tin OID hoặc thực hiện thao tác được yêu cầu.
* **SNMP Client (SNMPClientBenchmark.java, Test files)**:
  + Tạo các yêu cầu GET, WALK đến server.
  + Kiểm tra hiệu năng và tính chính xác của phản hồi.
* **Tiện ích và cấu hình (SnmpV3Util.java, Constants.java)**:
  + Hỗ trợ các tác vụ như quản lý bảo mật (SNMPv3), cấu hình địa chỉ IP, cổng, và OID.

#### ****Chi tiết triển khai****

1. **SNMP Client và Server**:
   * **SNMP Client**:
     + Gửi các yêu cầu GET, WALK đến server thông qua giao thức UDP.
     + Xử lý phản hồi và phân tích dữ liệu.
   * **SNMP Server**:
     + Lắng nghe yêu cầu trên cổng UDP.
     + Tra cứu giá trị OID từ một cơ sở dữ liệu giả lập hoặc cấu hình sẵn.
     + Trả lời với định dạng phù hợp cho từng phiên bản SNMP.
2. **Quản lý bảo mật trong SNMPv3**:
   * **Xác thực**: Sử dụng User-based Security Model (USM) để đảm bảo tính toàn vẹn và danh tính của yêu cầu.
   * **Mã hóa**: Bảo vệ dữ liệu khỏi bị nghe lén trong quá trình truyền tải.
   * **Quản lý người dùng**: Lưu trữ thông tin người dùng và mật khẩu để xác thực.
3. **Hỗ trợ các phiên bản SNMP khác nhau**:
   * **SNMPv1 và v2c**:
     + Sử dụng chung cơ chế đơn giản để gửi và nhận yêu cầu.
   * **SNMPv3**:
     + Tích hợp thêm các tính năng bảo mật như xác thực và mã hóa.

#### ****Các class/methods quan trọng****

1. **SNMPClientBenchmark**:
   * runBenchmark(): Gửi một lượng lớn yêu cầu đến server và ghi nhận hiệu năng.
   * measureResponseTime(): Tính toán thời gian phản hồi trung bình.
2. **SNMPServer**:
   * processRequest(): Xử lý các yêu cầu SNMPv1 và v2c.
   * sendResponse(): Gửi phản hồi đến client.
3. **SNMPServerV3**:
   * validateUser(): Xác thực thông tin người dùng.
   * encryptResponse(): Mã hóa dữ liệu trả về.
4. **SnmpV3Util**:
   * generateSecurityParams(): Tạo tham số bảo mật cho yêu cầu SNMPv3.
   * decryptMessage(): Giải mã thông điệp nhận được.
5. **CompareDifferentVersion**:
   * comparePerformance(): So sánh thời gian phản hồi giữa các phiên bản.
   * generateReport(): Tạo báo cáo phân tích hiệu năng.

#### ****Mô tả cách thực hiện benchmark****

Benchmark là một quá trình đo lường và đánh giá hiệu năng của hệ thống, cụ thể là các thành phần SNMP Client và Server trong ứng dụng. Để thực hiện benchmark, ứng dụng sử dụng class SNMPClientBenchmark.java, được thiết kế để đo thời gian phản hồi và hiệu suất khi thực hiện các yêu cầu SNMP.

**Quy trình thực hiện benchmark**:

1. **Chuẩn bị môi trường**:
   * Cấu hình SNMP Server để hỗ trợ các phiên bản SNMP (v1, v2c, v3).
   * Đảm bảo các thông số như địa chỉ IP, cổng, và thông tin xác thực (với SNMPv3) được thiết lập đúng.
   * Cài đặt các thiết bị giả lập nếu cần để mô phỏng môi trường thực tế.
2. **Thực hiện các yêu cầu SNMP**:
   * Gửi các yêu cầu GET hoặc WALK từ SNMP Client đến Server.
   * Lặp lại các yêu cầu với số lượng lớn để đo lường khả năng chịu tải.
   * Đo thời gian phản hồi của từng yêu cầu và tính toán trung bình.
3. **Đánh giá hiệu năng theo phiên bản**:
   * Chạy benchmark riêng biệt cho từng phiên bản SNMP (v1, v2c, v3).
   * So sánh thời gian phản hồi và số lượng yêu cầu xử lý mỗi giây (requests per second - RPS) giữa các phiên bản.
4. **Thu thập và phân tích dữ liệu**:
   * Ghi nhận thời gian phản hồi trung bình, tối đa, và tối thiểu.
   * Đánh giá hiệu suất dựa trên các yếu tố như bảo mật (SNMPv3) và khả năng xử lý đồng thời.

#### ****Kết quả benchmark****

Dưới đây là các kết quả dự kiến từ quá trình benchmark:

|  |  |  |  |
| --- | --- | --- | --- |
| **Phiên bản SNMP** | **Thời gian phản hồi trung bình (ms)** | **Số yêu cầu mỗi giây (RPS)** | **Nhận xét** |
| **SNMPv1** | 10 | 1000 | Hiệu năng tốt, nhưng không bảo mật |
| **SNMPv2c** | 12 | 950 | Hiệu năng tốt, hỗ trợ dữ liệu lớn hơn |
| **SNMPv3** | 30 | 500 | Bảo mật cao, nhưng hiệu suất giảm |

**Nhận xét chi tiết**:

1. **SNMPv1**:
   * Có thời gian phản hồi nhanh nhất do giao thức đơn giản.
   * Phù hợp cho các hệ thống yêu cầu hiệu năng cao nhưng không quan tâm đến bảo mật.
2. **SNMPv2c**:
   * Hiệu năng tương đối cao, gần với SNMPv1, nhưng hỗ trợ thêm các tính năng như Bulk Transfer, giúp giảm số lượng thông điệp cần thiết.
   * Đây là lựa chọn cân bằng giữa hiệu năng và khả năng quản lý dữ liệu lớn.
3. **SNMPv3**:
   * Hiệu suất giảm do yêu cầu xử lý bảo mật (xác thực và mã hóa).
   * Thích hợp cho các hệ thống yêu cầu bảo mật cao, như quản lý các thiết bị quan trọng hoặc trong môi trường doanh nghiệp lớn.