**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**



*BÁO CÁO ĐỒ ÁN CUỐI KỲ*

**Verification of Link-level Protocols**

**Môn: Cấu trúc dữ liệu và Giải thuật**

**GVHD: ThS. Huỳnh Xuân Phụng**

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SV** | **MSSV** |
| **1** | **Nguyễn Khang Thịnh** | **21110660** |
| **2** | **Trần Thành Lợi** | **21110536** |
| **3** | **Cao Đăng Duy** | **21110397** |

**TP. HCM, tháng 12 năm 2022**

**DANH SÁCH THÀNH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN SV | MSSV | Tỷ lệ hoàn thành |
| 1 | Nguyễn Khang Thịnh | 21110660 | 100% |
| 2 | Trần Thành Lợi | 21110536 | 100% |
| 3 | Cao Đăng Duy | 21110397 | 100% |

Mục lục

[POSTER 1](#_Toc123147377)

[PHẦN MỞ ĐẦU 2](#_Toc123147378)

[1. Lí do chọn bài báo 2](#_Toc123147379)

[2. Giới thiệu bài báo 2](#_Toc123147380)

[3. Mục tiêu 2](#_Toc123147381)

[4. Nội dung của báo cáo 2](#_Toc123147382)

[PHẦN NỘI DUNG 4](#_Toc123147383)

[1. Input, output và các biến cần nắm 4](#_Toc123147384)

[2. Kết luận của Stein Krogdahl về nghiên cứu 4](#_Toc123147385)

[3. Chứng minh công thức trong bài báo 5](#_Toc123147386)

[4. Trường hợp truyền dữ liệu < k phần tử 8](#_Toc123147387)

[5. Trường hợp vượt quá k phần tử 10](#_Toc123147388)

[6. Tổng quát hóa 11](#_Toc123147389)

[7. Phân tích hướng phát triển 11](#_Toc123147390)

[CÁC ỨNG DỤNG 13](#_Toc123147391)

[THU HOẠCH 14](#_Toc123147392)

# **POSTER**

Bài báo: Verification of Link-level Protocols

Tác giả: Stein Krogdahl

Graphical user interface, text, application

Description automatically generated

# **PHẦN MỞ ĐẦU**

## **1. Lí do chọn bài báo**

Mở đầu cuộc cách mạng công nghiệp 4.0, dữ liệu và thông tin đã đóng góp một vị trí quan trọng trong cuộc sống. Việc truyền và nhận dữ liệu trở thành một vấn đề nhận được nhiều sự quan tâm và nhiều nhà khoa học đã dày công nghiên cứu về chúng. Tiêu biểu trong số đó là Stein Krogdahl với bài báo ‘Verification of Link-level Protocols” được đăng trên tờ BIT 21 vào năm 1986.

## **2. Giới thiệu bài báo**

“Verification of Link-level Protocols” là một bài báo nghiên cứu về các đặc tính của các mô hình truyền dữ liệu phổ biến. Stein Krogdahl muốn chứng minh sự đúng một phần của khung giao thức (truyền – nhận dữ liệu). Ông đã đặt ra một vài giả thiết, kiểm chứng chúng thông qua mô hình Alice-Bill được đề cập trong bài báo. Ông đã tìm ra các mối quan hệ bất biến, chúng duy trì trong suốt quá trình khảo sát mô hình. Từ đó đúc kết ra kết luận và hoàn thành công trình nghiên cứu.

## **3. Mục tiêu**

Bài báo cáo của nhóm em được thực hiện với mục tiêu:

- Khảo sát mô hình Alice-Bill mà Stein Krogdahl đã đặt ra.

- Kiểm chứng các mối quan hệ bất biến (invariants), các giả thiết (lemmas) và định lý (theorem) trong bài báo.

- Ứng dụng các kiến thức trong môn Cấu trúc dữ liệu và Giải thuật, đặc biệt là kiến thức về Linked-list, Queue và khả năng phân tích thuật toán.

- Có khả năng viết lại chương trình Alice-Bill bằng ngôn ngữ C/C++.

## **4. Nội dung của báo cáo**

1. Input, output và các biến cần nắm

2. Kết luận của Stein Krogdahl về nghiên cứu

3. Trường hợp truyền dữ liệu < k phần tử

4. Trường hợp truyền dữ liệu > k phần tử

5. Tổng quát hóa

6. Thu hoạch

# **PHẦN NỘI DUNG**

## **1. Input, output và các biến cần nắm**

**1.1. Input, output**

- Input: Message queue và Acknowledgment queue (hàng đợi tin nhắn và hàng đợi thông báo).

- Ouput: True/False của các giả thiết (lemma) và định lý (theorem).

**1.2. Các biến cần nắm**

Trong mô hình Alice-Bill mà Stein Krogdahl đặt ra, chúng ta có hai hàng đợi là Message queue và Acknowledgment queue.

Khi Alice gửi tin nhắn, cô sẽ chèn 1 tin vào phía sau của Message queue. Khi Bill nhận được, anh sẽ xóa nó khỏi hàng đợi. Tương tự với việc gửi thông báo của Bill.

A: biến cục bộ do Alice giữ. Đây là số lượng tin nhắn mà Bill thông báo với Alice rằng anh ta đã nhận được.

j: mỗi tin nhắn do Alice gửi sẽ đính kèm vị trí là j.

B: biến cục bộ do Bill giữ. Đây là số lượng tin nhắn mà Bill đã nhận được.

jmax: đây là vị trí lớn nhất (tin nhắn sau cùng) bị Bill xóa khỏi hàng đợi tin nhắn.

## **2. Kết luận của Stein Krogdahl về nghiên cứu**

Như đã trình bày, Stein Krogdahl muốn chứng minh tính hợp lệ của các mô hình truyền dữ liệu mà ông đưa ra. Ông làm điều đó bằng cách thiết lập một khung giao thức và chứng minh các giả thiết ông đặt ra.

Để thực hiện, ông đưa ra một mô hình đơn giản, trong đó:

Alice gửi các tin nhắn cho Bill. Bill gửi lại các thông báo “đã đọc” cho Alice. Nhưng do đường truyền đông đúc (noisy), nên chỉ có thể gửi một lượng thông tin nhất định trong mỗi tin nhắn và thông báo.

Stein Krogdahl cố gắng tìm ra các quy luật, các mối quan hệ trong quá trình truyền dữ liệu, Ông cũng đặt ra câu hỏi rằng nếu số lượng tin nhắn vượt quá giới hạn thì lượng dữ liệu đi kèm mỗi tin nhắn / thông báo sẽ thay đổi như thế nào?

Ông đưa ra kết luận:

*“Only a limited amount of information about j needs to appear in the message queue, and only a limited amount of information about b needs to appear in the acknowledgment queue.”*

*“Chỉ một lượng thông tin giới hạn về j cần xuất hiện trong hàng đợi tin nhắn, và chỉ một lượng thông tin nhất định về b cần xuất hiện trong hàng đợi thông báo.”*

## **3. Chứng minh công thức trong bài báo**

**3.1. Lemma 1**

Let the contents of the acknowledgment queue be

b1 … br

from the front to the rear, where r 0. Then

A b1  … br B.

*Prove the stated relation remains invariant.*

Chart

Description automatically generated

Chứng minh:

Khởi tạo A = B = 0.

Khi r = 0, biểu thức 1 trở thành: A B (luôn đúng, do A nhận giá trị từ B, vậy nên A sẽ luôn nhỏ hơn hoặc bằng B).

Khi r > 0:

* Ở hoạt động A2, Alice sẽ nhận một thông báo ‘b’ và gán A b. Do đó biến A sẽ luôn trễ hơn b một hoạt động, hay nói cách khác là Ab1.
* Biểu thức ‘b1 br’ là điều hiển nhiên đúng, vì các thông báo được gửi từ Bill với số lượng tin nhắn mà Bill đọc được chắc chắn sẽ là tăng dần.
* Có thể thấy, biến B sẽ tăng lên ngay khi Bill nhận được một tin nhắn, và sau đó anh mới gửi thông báo br với br = B. Vậy nên br sẽ trễ hơn B một hoạt động, hay nói cách khác là br luôn nhỏ hơn hoặc bằng B Bbr.

Vì vậy, khi r > 0: A ≤ b1 ≤ … ≤ br ≤ B

Tóm lại:

*The stated relation remains invariant under four basic operations A1, A2, B1, B2.*

*Mối quan hệ kể trên là bất biến trong suốt quá trình thực hiện các hoạt động A1, A2, B1, B2.*

**3.2. Lemma 2**

Let the contents of the message queue be

Mj1 … Mjr

from the front to the rear, where r 0, and let jmax be the maximum index of any message that has ever been removed from the message queue. (If nothing has ever been removed, let jmax = 0). Let j0 = jmax and jr+1 = A. Then

ji < ji’ + k for 0 i i' r + 1

*Prove the stated relation remains invariant.*

Chart, scatter chart

Description automatically generated

Chứng minh:

Khi r = 0, chúng ta có i = 0, i’ = 1

0 = j0 < j1 + k = A + k = k hay 0 < k đúng.

Khi r > 0:

* Hoạt động A1 push một tin nhắn mới ‘j’ or ‘jr+1’ vào hàng đợi tin nhắn (A j < A + k), khi này hàng đợi sẽ trở thành: j1 … jr jr+1

Từ đề bài, chúng ta có jr+1 = A. Vì vậy biểu thức trên sẽ trở thành: jr+1 < jr+2 + k hoặc A < 0 + k đúng.

* Hoạt động A2 không làm giảm biến A, bởi vì biến B luôn tăng, mà ở hoạt động A2, ta gán A = b (với b = B) vậy nên biến A chỉ có thể tăng.
* Hoạt động B1 không ảnh hưởng tới biểu thức.
* Từ đề bài, chúng ta có jmax = 0. Hoạt động B2 sẽ thay thế jmax bởi max(jmax , j1)

do đó biểu thức trên không đổi.

Vì vậy, khi r > 0: ji < ji’ + k for 0 ≤ i < i' ≤ r + 1.

Tóm lại:

*The stated relation remains invariant under four basic operations A1, A2, B1, B2.*

*Mối quan hệ kể trên là bất biến trong suốt quá trình thực hiện các hoạt động A1, A2, B1, B2.*

## **4. Trường hợp truyền dữ liệu < k phần tử**

k là kích thước kho lưu trữ đệm do Alice quản lý. Giả sử ban đầu k = 5.

*Alice gửi tin nhắn đầu tiên* ***(hoạt động A1).***

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

*Bill nhận tin nhắn đó và xóa nó khỏi hàng đợi. Biến B tăng lên 1 (Bill đã đọc 1 tin nhắn), và biến jmax được gán bằng 0 (vị trí tin nhắn lớn nhất mà Bill đã đọc).* ***(hoạt động B2).***

*Bill gửi một thông báo cho Alice kèm số lượng tin nhắn mình đã đọc với b = B = 1.* ***(hoạt động B1).***

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

*Tiếp theo, Alice sẽ nhận được thông báo này, xóa nó khỏi hàng đợi và biến A được cập nhật A = b = 1.* ***(hoạt động A2).***

Graphical user interface, text, application, chat or text message, Teams

Description automatically generated

Tương tự, sau khi gửi tiếp 4 tin nhắn còn lại, B sẽ tăng lên 5, jmax tăng lên 4. Nếu lúc này Bill gửi thông báo cho Alice, biến A sẽ được cập nhật bằng 5.

## **5. Trường hợp vượt quá k phần tử**

Trong trường hợp này, chúng ta sẽ gửi 6 tin nhắn. Việc gửi và nhận tin nhắn / thông báo vẫn diễn ra tương tự, nhưng chúng ta sẽ không gửi thông báo với b = B nữa. Stein Krogdahl quan sát và nhận ra rằng khi trao đổi quá k tin nhắn, Bill chỉ cần gửi một phần dư vị trí của tin nhắn, Alice có thể tự hiểu và suy ra số lượng thực sự mà Bill đã đọc.

Graphical user interface, text, application

Description automatically generated

Việc Alice gửi 6 tin nhắn (>k) và Bill gửi lại thông báo với b = 1, Alice có thể tự suy ra rằng Bill đã nhận đủ 5 tin nhắn đầu tiên và 1 tin nhắn sau cùng.

Điều này dẫn chúng ta đến kết luận nêu trên của Stein Krogdahl:

*“Only a limited amount of information about j needs to appear in the message queue, and only a limited amount of information about b needs to appear in the acknowledgment queue.”*

*“Chỉ một lượng thông tin giới hạn về j cần xuất hiện trong hàng đợi tin nhắn, và chỉ một lượng thông tin nhất định về b cần xuất hiện trong hàng đợi thông báo.”*

## **6. Tổng quát hóa**

Sau khi đúc kết được kết luận trên, Stein Krogdahl nhận ra rằng định lý đó cũng có thể được áp dụng cho trường hợp mà hàng đợi không vận hành theo quy tắc first-in-first-out, miễn là nó đáp ứng được một số quy luật nhất định về hoán vị.

Một bài báo tương tự của N. V. Stenning cũng chỉ ra trường hợp này, nó yêu cầu ứng với mỗi tin nhắn được gửi đi phải nhận được trở lại một thông báo. Đáp ứng hai điều kiện trên, định lý về việc truyền dữ liệu của Stein Krogdahl có thể được duy trì với bất kỳ hàng đợi nào, dù cho chúng không hoạt động theo cách first-in-first-out.

## **7. Phân tích hướng phát triển**

Như Stein Krogdahl đề cập, trong thực tế, Alice là một chương trình hệ thống dùng để nhận tin nhắn một cách tuần tự từ người dùng. Còn Bill là một chương trình hệ thống dùng để vận chuyển tin nhắn một cách tuần tự đến người dùng khác.

Mô hình truyền dữ liệu được sử dụng trong bài báo là một chiều (một người gửi tin nhắn – một người gửi thông báo). Chúng ta có thể mở rộng nó sang hai chiều, tức là mỗi người đều có thể gửi / nhận tin nhắn và thông báo như nhau.

Vậy chúng ta làm thế nào?

*Chúng ta sẽ tạo ra một middleware để truyền tin nhắn và thông báo*

7.1. Middleware là gì?

Middleware là một chương trình trung gian, nó là những đoạn mã nằm giữa các request và response. Nó sẽ nhận lệnh từ request, thực thi các lệnh đó và trả về kết quả thông qua response.

7.2. Kết hợp giữa Alice và Bill

Chương trình trung gian của chúng ta sẽ là sự kết hợp giữa Alice và Bill. Một chương trình có thể nhận tin nhắn từ người dùng A, gửi chúng tới người dùng B, và theo chiều ngược lại.

Các hàng đợi tin nhắn (message queue) và hàng đợi thông báo (acknowledgment queue) sẽ được quản lý bởi chương trình trung gian này.

7.3. Tạo định nghĩa mới cho user

Giờ đây, Alice không chỉ còn là người gửi tin nhắn, Bill không chỉ là người gửi thông báo. Chúng ta cần một định nghĩa mới về user. Một user với đầy đủ chức năng gửi / nhận tin nhắn và thông báo.

7.4. Áp dụng các quy tắc truyền dữ liệu của Stein Krogdahl

Middleware mà chúng ta xây dựng vẫn sẽ giữ nguyên các tính chất như Lemma 1, 2 và Theorem mà Stein Krogdahl đã chứng minh.

Sau khi xây dựng thành công middleware, chúng ta sẽ có một chương trình trung gian hoàn thiện, có thể gửi và nhận tin nhắn theo 2 chiều.

# **CÁC ỨNG DỤNG**

**1. Count garbled messages** (Đếm số lượng tin nhắn bị hỏng)

**2. Check if transmission line is broken** (Kiểm tra đường truyền có bị lỗi hay không)

**3. Message oversend check** (Kiểm tra người dùng có gửi nhiều hơn số lượng tin nhắn cho phép hay không)

# **THU HOẠCH**

Sau khi nghiên cứu bài báo của Stein Krogdahl, nhóm em đã thu hoạch được một số nội dung như sau:

*Một là*, ôn tập lại kiến thức về tiếng Anh, tăng cường khả năng dịch thuật và khả năng phân tích bài báo.

*Hai là*, vận dụng kiến thức đã học từ môn Cấu trúc dữ liệu và Giải thuật để viết thuật toán, giải thích công thức trong bài báo. Vận dụng kiến thức về Queue và Linked-list để viết code mô phỏng chương trình.

*Ba là*, áp dụng được những kiến thức đã học vào thực tế, tìm được các ứng dụng và viết code mô phỏng cho các ứng dụng đó.