

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

\*

# ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ XUẤT GIAO THỨC ĐỊNH TUYẾN TRÁNH  
HỒ SỬ DỤNG HỌC TĂNG CƯỜNG TRONG  
MẠNG CẢM BIẾN KHÔNG DÂY

Sinh viên thực hiện : **Lê Thị Khanh**  
Lớp Việt Nhật IS - K59

Giáo viên hướng dẫn : **TS.**  
**Nguyễn Thanh Hùng**

Hà Nội, 05-2019

# LỜI CẢM ƠN

Khi những dòng cuối cùng của đề án này được hoàn thành cũng là lúc em biết mình nên bày tỏ lòng biết ơn tới gia đình, thầy cô, bạn bè - những người đã luôn ở bên cạnh ủng hộ và cổ vũ em trong suốt thời gian qua.

Trước tiên em xin được dành lời biết ơn sâu sắc nhất tới cô giáo, TS. **Nguyễn Phi Lê** - người đã luôn tận tình dẫn dắt em từ những bước đầu tiên đến với con đường nghiên cứu khoa học. Ở cô, em thấy được sự nhiệt huyết, tận tụy và cống hiến của một người làm nghiên cứu. Cô là một *người thầy* mà mọi vấn đề khó khăn trong nghiên cứu của em đều có thể được giải quyết khi thảo luận cùng cô, không phải bằng cách áp đặt suy nghĩ của cô lên em mà bằng cách hướng em tới những tư duy học thuật đúng đắn, giúp em tự do khám phá và phát triển bản thân. Em cảm thấy mình may mắn vì đã gặp được một *người thầy - người chị* như cô.

Lời tiếp theo, em xin chân thành cảm ơn TS. **Nguyễn Thanh Hùng** đã luôn hướng dẫn, chỉ bảo và giúp đỡ em trong suốt quá trình làm đề án. Sự quan tâm và theo dõi sát sao của thầy đã giúp em có thêm nhiều động lực để hoàn thành tốt đề án này.

Người thầy cuối cùng mà em muốn bày tỏ lòng biết ơn chân thành, đó là thầy giáo, PGS.TS **Nguyễn Khanh Văn**. Tuy không trực tiếp hướng dẫn em, nhưng thầy đã cho em những lời khuyên vô cùng quý giá về phương hướng giải quyết vấn đề trong quá trình thực hiện đề án.

Xin cảm ơn tất cả những người bạn của tôi ở trường Đại học Bách Khoa Hà Nội thân yêu, những người đã cho tôi những nụ cười, những kỷ ức thân yêu trong những năm tháng vừa qua.

Đặc biệt, tôi muốn cảm ơn **Chính, Hoàng, Nga, Việt** - những người bạn đã kề vai sát cánh cùng tôi trong suốt 5 năm đại học nhiều niềm vui nhưng cũng không thiếu những lúc khó khăn, những người bạn đã kéo tôi ra khỏi những giờ học mải miết để tận hưởng giây phút quý giá của tuổi thanh xuân. Họ là những người bạn mà tôi luôn trân trọng.

Lời cuối cùng xin dành cho những người quan trọng nhất trong cuộc đời tôi. Dù có đi hết cả cuộc đời, có lẽ tôi cũng không thể trả hết tình yêu và sự chăm sóc mà gia đình đã dành cho tôi. Con xin cảm ơn gia đình đã luôn ở bên làm điểm tựa tinh thần vững chắc giúp con vượt qua khó khăn không chỉ trong quá trình thực hiện đề án mà còn trên toàn bộ quá trình học tập, rèn luyện tại trường Đại học Bách khoa Hà Nội!

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

## 1. Thông tin về sinh viên

Họ và tên sinh viên: Lê Thị Khanh

Điện thoại liên lạc: 0948 330 645

Email: lekhanh.lamson@gmail.com

Lớp: Việt Nhật IS - K59

Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: Trường ĐHBKHN

Thời gian làm DATN: Từ ngày 16/10/2018 đến 24/05/2019

## 2. Mục đích nội dung của DATN

Đề xuất giao thức định tuyến tránh hố trong mạng cảm biến không dây sử dụng phương pháp Q-learning để đạt được hiệu quả cao về 3 yếu tố: tỷ lệ gói tin được truyền thành công, thời gian đi từ nguồn tới đích của gói tin, và thời gian sống của mạng.

## 3. Các nhiệm vụ cụ thể của DATN

- Nghiên cứu các giao thức định tuyến địa lý trong mạng cảm biến không dây đã được sử dụng trước đây.
- Nghiên cứu về Q-learning và các giao thức định tuyến sử dụng Q-learning trong mạng cảm biến không dây.
- Xây dựng giao thức định tuyến tránh hố sử dụng Q-learning trong mạng cảm biến không dây.
- Cài đặt giao thức đề xuất và các giao thức so sánh vào công cụ mô phỏng NS2.

## 4. Lời cam đoan của sinh viên

Tôi - *Lê Thị Khanh* - cam kết DATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thanh Hùng*.

Các kết quả nêu trong DATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày      tháng      năm 2019

Tác giả DATN

*Lê Thị Khanh*

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN  
và cho phép bảo vệ

*Hà Nội, ngày      tháng      năm 2019*  
Giáo viên hướng dẫn

*TS. Nguyễn Thanh Hùng*

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong cuộc sống hiện đại, thay vì phải đối mặt với nạn đói, bệnh dịch, chiến tranh, chúng ta đang phải đối mặt với ô nhiễm môi trường và biến đổi khí hậu. Các kết luận về mức độ ô nhiễm không khí, hay sự thay đổi nồng độ khí nhà kính,... được đưa ra là nhờ một phần vào các trạm quan sát môi trường được triển khai từ mạng cảm biến không dây (Wireless Sensor Networks). Có thể nói các hệ thống mạng cảm biến không dây không hiện hữu trực tiếp và thân thuộc với chúng ta trong cuộc sống hằng ngày, nhưng chúng đang góp phần vào xây dựng và cải thiện cơ sở hạ tầng sống của con người. Bởi đặc điểm về bán kính truyền tin hẹp và tài nguyên hạn chế của các mạng cảm biến không dây, chúng cần một giao thức truyền tin nhanh chóng, đơn giản, và chính xác.

Trong khi các giao thức truyền tin trong trường hợp mạng bằng phẳng, nút mạng phân bố đều, không có vật cản đã đạt được kết quả khá tốt, thì các giao thức truyền tin trong trường hợp mạng có vật cản (như hồ nước, phương tiện giao thông...) tuy được đề xuất khá nhiều nhưng vẫn còn tồn tại các nhược điểm về: độ phức tạp tính toán, lượng thông tin cần lưu trữ và phát tán...

Trong đồ án này, chúng tôi sẽ đề xuất một giao thức truyền tin tránh vật cản (tránh hố) trong mạng cảm biến không dây - giao thức ***HYBRID***, là sự kết hợp giữa định tuyến địa lý và phương pháp học tăng cường, nhằm giải quyết các vấn đề còn tồn đọng về độ phức tạp tính toán, lượng thông tin cần lưu trữ và phát tán... của các giao thức định tuyến tránh hố đã được đề xuất trước đây.

Đồ án này bao gồm các phần: Nghiên cứu và phân tích các giao thức định tuyến tránh hố trước đây; Lý luận phương pháp xây dựng và đề xuất giao thức HYBRID; Trình bày kết quả thí nghiệm đánh giá hiệu năng của giao thức HYBRID; Kết luận và nêu hướng phát triển trong tương lai.

## ABSTRACT OF THESIS

Recent years have witnessed the emergence of wireless sensor networks (WSNs) which consist of tiny sensor nodes deployed over a region of interest to monitor and control the physical environment. Wireless sensor networks have been widely used in various domains such as military target tracking and surveillance, natural disaster relief, agricultural and environmental monitoring, biomedical health monitoring, etc. In this thesis, we focus on large-scale WSNs which are based on the cooperation of a large number of sensor nodes. Typically, large-scale WSNs are used in monitoring applications such as agricultural monitoring, climate monitoring, forest monitoring, weather monitoring, etc.

Due to the short-range communication nature of sensor nodes, routing becomes one of the most important issues which has received intensive research attention. As sensor nodes are equipped with only limited and non-rechargeable batteries, conserving energy consumption is an important factor in designing routing protocol. Energy conservation can be achieved by shortening the routing path and reducing the overhead caused by control packets. In many applications, the network can't achieve its objective if all the nodes can't sense or report the sensed data, thus the death of even only one node may cause the network to operate un-functionally. Accordingly, balancing traffic over the network to extend the network lifetime is another important designing factor of routing protocol in WSNs.

Geographic routing has been widely used in wireless sensor networks because of its simplicity and efficiency. However, when subjected to routing holes (i.e., regions without sensor nodes that have communication capability), geographic routing suffers from the so called local minimum phenomenons. Many schemes were proposed to resolve these problems but all of the existing protocols cannot solve the load imbalance and routing path enlargement problems thoroughly.

In this thesis, we propose a novel protocol for bypassing hole in wireless sensor networks which can prolong the network's lifetime, ensure the successful packet delivery ratio, and shorten the end-to-end delay of data packet. The key ideas behind our approaches are the combination between geographic routing and machine learning (Q-learning in detail). Geographic routing method will help data packets escape/bypass holes, while, machine learning could help sensor node to choose appropriate next hop in order to optimize the routing efficiency.

# Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>8</b>
1.1	Mạng cảm biến không dây . . . . .	8
1.2	Định tuyến địa lý và vấn đề hố định tuyến . . . . .	9
1.2.1	Định tuyến địa lý . . . . .	9
1.2.2	Hố định tuyến . . . . .	10
1.3	Các hướng giải quyết bài toán định tuyến tránh hố . . . . .	10
1.4	Phương hướng và mục tiêu nghiên cứu của đề án . . . . .	11
1.5	Các kết quả đạt được . . . . .	12
1.6	Cấu trúc đề án . . . . .	12
<b>2</b>	<b>Các giao thức định tuyến tránh hố trong mạng cảm biến không dây</b>	<b>13</b>
2.1	Giao thức xác định hố . . . . .	13
2.2	Giao thức định tuyến tránh hố . . . . .	15
2.2.1	Tiếp cận theo hướng đi men biên hố . . . . .	15
2.2.2	Tiếp cận theo hướng xây dựng vùng cấm xung quanh hố . . . . .	15
2.2.3	Tiếp cận theo hướng heuristic . . . . .	17
<b>3</b>	<b>Phương pháp học tăng cường</b>	<b>19</b>
3.1	Lý thuyết và ứng dụng của phương pháp học tăng cường . . . . .	19
3.2	Giới thiệu các thuật toán học tăng cường . . . . .	20
3.3	Thuật toán Q-learning . . . . .	20
3.4	Các giao thức định tuyến sử dụng học tăng cường trong mạng cảm biến không dây . . . . .	21
<b>4</b>	<b>Đề xuất giao thức định tuyến tránh hố trong mạng cảm biến không dây sử dụng phương pháp học tăng cường</b>	<b>23</b>
4.1	Giả thiết bài toán . . . . .	23
4.1.1	Mô hình mạng . . . . .	24
4.1.2	Các ký hiệu và định nghĩa . . . . .	24
4.2	Tổng quan . . . . .	25
4.3	Giao thức xác định hố . . . . .	26
4.4	Giao thức phát tán thông tin vùng lõm của hố . . . . .	26
4.5	Giao thức truyền tin . . . . .	28
4.5.1	Các cơ chế truyền tin . . . . .	28

4.5.2	Sử dụng Q-learning để tối ưu hóa đường định tuyến của gói tin	34
<b>5</b>	<b>Thí nghiệm và đánh giá hiệu năng của giao thức đề xuất</b>	<b>38</b>
5.1	Kịch bản thí nghiệm	38
5.2	Các tiêu chí đánh giá hiệu năng	39
5.3	Phân tích kết quả thí nghiệm	40
5.3.1	Thời gian sống của mạng	40
5.3.2	Thời gian gói tin đi từ nguồn tới đích	41
5.3.3	Tỷ lệ gói tin tới đích thành công	42
5.3.4	Năng lượng trung bình để gửi một gói tin	43
5.3.5	Tổng reward trung bình theo thời gian của các gói tin trong giao thức HYBRID	44
<b>6</b>	<b>Kết luận</b>	<b>45</b>
6.1	Kết quả đạt được	45
6.2	Hướng phát triển	45

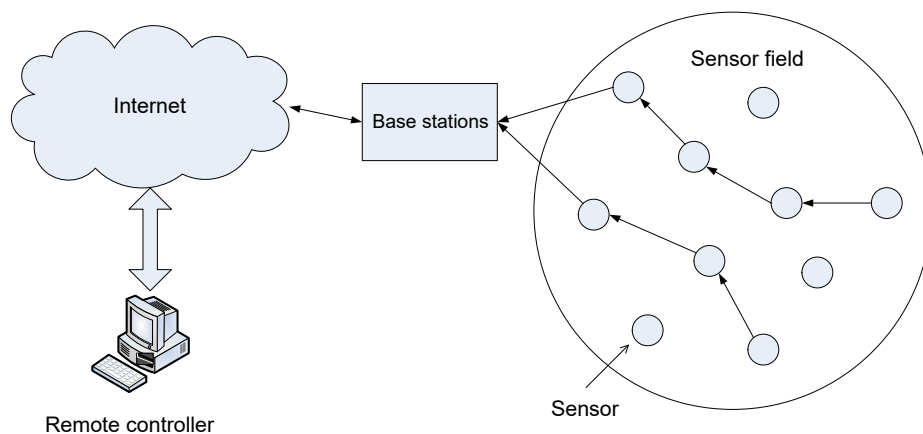


# Chương 1

## Giới thiệu

### 1.1 Mạng cảm biến không dây

Mạng cảm biến không dây là một tập hợp gồm nhiều nút cảm biến (từ vài trăm tới vài nghìn nút) có thể kết nối không dây với nhau, được triển khai trong một vùng để thu thập thông tin của đối tượng cần khảo sát. Một nút cảm biến (còn được gọi là *bộ cảm biến*, hay *cảm biến*) là một thiết bị điện tử có khả năng cảm nhận những trạng thái hay quá trình vật lý hay hóa học từ môi trường, và biến đổi thành tín hiệu điện để thu thập thông tin về trạng thái hay quá trình đó (theo wikipedia [1]). Thông tin này sau đó sẽ được truyền về trạm (base station) để xử lý dữ liệu. Bán kính cảm biến của một cảm biến khá thấp, thường không vượt quá 10m, và đồng thời bán kính truyền tin của nó cũng chỉ nằm trong khoảng từ vài chục tới vài trăm mét.



Hình 1.1: Mạng cảm biến không dây

Mạng cảm biến không dây được ứng dụng khá rộng rãi trong thực tế: theo dõi mục tiêu trong quân sự [2, 3], cứu trợ thiên tai [4, 5, 6, 7, 8], giám sát trong nông nghiệp và môi trường [9, 10, 11, 12, 13], theo dõi sức khỏe [14, 15, 16], etc.

Trong đề án này, chúng tôi tập trung vào mạng cảm biến không dây có quy mô lớn (cỡ vài nghìn nút cảm biến). Chúng ta có thể bắt gặp các mô hình mạng cảm biến không dây quy mô lớn này trong các ứng dụng về giám sát trên diện rộng như: giám sát xâm

nhập bất hợp pháp trong quân sự, đo độ ô nhiễm không khí, giám sát giao thông, đo độ ô nhiễm của sông hồ, etc.

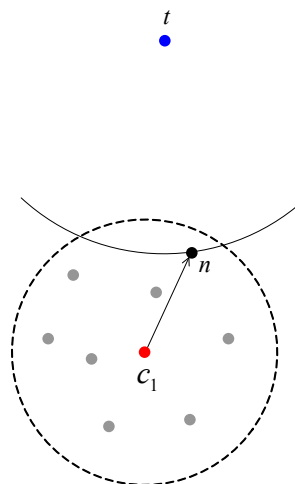
Bắt đầu từ phần này chúng tôi sẽ dùng các thuật ngữ *cảm biến*, *nút cảm biến*, *nút mạng*, *node*, hoặc *bộ cảm biến* để chỉ một nút cảm biến trong mạng, và thuật ngữ *hàng xóm*, hay *1-hop neighbor* của một nút để chỉ những nút nằm trong bán kính truyền tin của nút đó.

## 1.2 Định tuyến địa lý và vấn đề hồ định tuyến

### 1.2.1 Định tuyến địa lý

Như đã nêu ở trên, sau khi các cảm biến thu thập thông tin từ môi trường, chúng cần truyền dữ liệu về trạm để xử lý. Với ba đặc trưng cơ bản của cảm biến là khả năng tính toán thấp, ít bộ nhớ và năng lượng không bền (theo [9], một cảm biến có không quá 640Kb bộ nhớ, và theo [9, 17], hầu hết các cảm biến đều có năng lượng không vượt quá 5V); việc chọn một giao thức định tuyến đơn giản và phù hợp để duy trì cho mạng hoạt động chính xác trong thời gian dài là vô cùng quan trọng.

Định tuyến địa lý là một trong những phương pháp định tuyến phổ biến nhất trong mạng cảm biến không dây bởi nó có tính đơn giản và hiệu quả. Định tuyến địa lý là giao thức định tuyến sử dụng thông tin tọa độ và ID của *1-hop neighbor* (nút nằm trong bán kính truyền tin) để định tuyến, thay vì dùng bảng định tuyến như các mạng có dây hay mạng wifi. Một trong những giao thức định tuyến cổ điển đơn giản và phổ biến nhất trong nhóm giao thức định tuyến địa lý đó là *giao thức GREEDY*, là giao thức mà gói tin luôn được gửi tới hàng xóm có khoảng cách tới đích gần nhất trong số tất cả các hàng xóm của nút hiện tại.



Hình 1.2: Định tuyến địa lý.  $t$  là nút đích, nút hiện tại  $c_1$  chọn hàng xóm  $n$  gần  $t$  nhất để gửi gói tin tới.

### 1.2.2 Hố định tuyến

**Hố định tuyến** là vùng không tồn tại nút cảm biến, hoặc tồn tại nút cảm biến nhưng chúng bị chết khiến gói tin không thể đi qua được.

Giao thức greedy hoạt động khá tốt cho những mạng có mật độ nút mạng cao, tuy nhiên trong trường hợp mạng có mật độ thưa, hay có sự xuất hiện của hố định tuyến, giao thức này gặp phải hiện tượng **cực tiểu địa phương** - là hiện tượng một nút mạng ở vị trí gần đích hơn tất cả các hàng xóm của nó. Lúc này giao thức greedy sẽ không truyền tin tiếp được nữa, và nút mạng đó được gọi là *stuck node*.

Trong thực tế hố định tuyến xuất hiện khi mạng cảm biến không đầy được triển khai ở những vùng có vật cản như ao, hồ,...; hoặc khi các nút mạng bị chết do: cạn kiệt năng lượng, cháy rừng, bị va chạm với sinh vật hay phương tiện khác dẫn tới bị hỏng, etc.

Nhìn chung, có hai hướng tiếp cận để giải quyết vấn đề hố định tuyến đó là: 1) đề xuất các phương pháp định tuyến tránh hố; 2) đưa sensor tới để lấp đầy hố định tuyến. Tuy nhiên thì phương pháp thứ 2 có phần tốn kém chi phí và trong trường hợp mạng gặp vật cản thì việc đặt sensor ở những vùng này cũng khó khăn hơn. Có khá nhiều giao thức truyền tin trong mạng có hố định tuyến (còn gọi là giao thức định tuyến tránh hố) đã được đề xuất, và chúng ta sẽ thảo luận ở chương 1.3 ngay tiếp theo đây.

Từ các chương sau, hố định tuyến sẽ được gọi tắt là *hố*.

## 1.3 Các hướng giải quyết bài toán định tuyến tránh hố

Có bốn cách tiếp cận phổ biến để giải quyết bài toán định tuyến tránh hố đó là:

- **Tiếp cận theo hướng đi men theo biên hố:** gói tin sẽ được truyền đi theo giao thức greedy cho tới khi gặp hố. Khi gặp hố, gói tin sẽ đi men theo các nút nằm trên biên hố để tránh hố và đi tới đích.
- **Tiếp cận theo hướng xây dựng vùng cấm xung quanh hố** ([33, 34, 32, 35, 19]: trong các trường hợp đoạn thẳng nối nguồn và đích cắt hố, cách tiếp cận theo hướng đi men biên hố sẽ gặp phải hai tình trạng: các nút trên biên hố phải làm việc liên tục dẫn tới cạn kiệt năng lượng và chết đi, khiến hố bị lan rộng ra; và trong trường hợp biên hố rộng, nguồn nghèo phức tạp thì việc đi men biên hố sẽ kéo dài đường đi của gói tin, đồng thời làm tăng thời gian truyền tin từ nguồn tới đích, cũng làm tốn kém năng lượng của nhiều nút mạng hơn. Vì vậy nhiều cơ chế định tuyến dựa theo đa giác bao phủ quanh hố đã được đưa ra. Ý tưởng chung đó là dùng các hình dạng đa giác đơn giản như hình tròn, hình elip, hình lục giác, bát giác để bao phủ hố và trở thành vùng cấm. Thông tin về vùng cấm này sẽ được phát tán đi để những nút nhận được thông tin sẽ xây dựng đường định tuyến tránh hố. Ý tưởng này vừa rút ngắn đường đi của gói tin trong một số trường hợp, vừa tránh được tình trạng hố bị lan rộng.

- **Tiếp cận theo hướng dựa vào kinh nghiệm định tuyến trong quá khứ** (heuristic routing) ([20]): cơ chế định tuyến theo hướng này sẽ dùng thông tin định tuyến trong quá khứ để "học" về mô hình mạng, xây dựng các hàm để tối ưu hiệu quả truyền tin và xây dựng một số thông tin điều khiển cần thiết cho việc định tuyến ở các vùng quan trọng như khu vực xung quanh biên hố. Hướng tiếp cận này không cần biết trước thông tin về hố.

Như vậy, phương pháp định tuyến men theo biên hố sẽ gây ra tình trạng hố lan rộng và đường định tuyến bị kéo dài.

Phương pháp định tuyến dựa trên đa giác bao phủ đã được đưa ra để giải quyết hai vấn đề này. Hướng tiếp cận này đã được chứng minh là khá hiệu quả trong việc đảm bảo cân bằng tải, kéo dài thời gian sống cho mạng và đảm bảo được cận trên của độ dài đường định tuyến, tuy nhiên lại không giải quyết trường hợp nút nguồn nằm trong vùng lõm của hố. Phương pháp định tuyến dựa trên các đường vị tự biên hố cũng cho hiệu quả rất tốt và giải quyết cả trường hợp nút nguồn nằm trong vùng lõm của hố. Tuy nhiên độ phức tạp tính toán đang còn cao dẫn tới thời gian gửi tin từ nguồn tới đích bị kéo dài.

Phương pháp định tuyến heuristic không cần tốn nhiều năng lượng trong việc phát hiện hố và xây dựng các thông tin điều khiển, nhưng cũng vì đó mà các giao thức định tuyến heuristic tính đến thời điểm hiện tại chưa đạt hiệu quả cao. Gần đây, một số phương pháp định tuyến heuristic sử dụng học máy đã được đưa ra tuy nhiên chúng đều chưa giải quyết được vấn đề hố định tuyến.

Chi tiết về các hướng tiếp cận trên sẽ được trình bày ở chương 2.2.

## 1.4 Phương hướng và mục tiêu nghiên cứu của đề án

Dựa vào các nhận xét đã nêu ra ở chương 1.3, mục tiêu của chúng tôi là xây dựng một cơ chế định tuyến tránh hố đáp ứng các điều kiện:

1. Định tuyến được trong hai trường hợp nút nguồn nằm trong và ngoài vùng lõm của hố.
2. Hiệu quả thuật toán cao. Với hiệu quả của thuật toán được đánh giá bởi 3 tiêu chí: *thời gian sống của mạng, thời gian truyền tin từ nguồn tới đích, tỉ lệ số gói tin được gửi thành công.*

Để đáp ứng được hai điều kiện trên, chúng tôi đã chọn cách tiếp cận kết hợp giữa định tuyến địa lý và phương pháp học tăng cường, cụ thể là thuật toán Q-learning. Trong đó, định tuyến địa lý sẽ đảm bảo gói tin đi theo đúng hướng tới đích và tránh được

hồ, còn phương pháp học tăng cường nhằm mục tiêu đạt hiệu quả cao trong việc chọn đường đi tối ưu, tính toán nhẹ, và tiết kiệm bộ nhớ cho cảm biến.

## 1.5 Các kết quả đạt được

Sau quá trình nghiên cứu và thực hiện, đề án đã đạt được những kết quả như sau:

- Đã nghiên cứu, kế thừa ưu điểm và giải quyết thiếu sót của các giao thức định tuyến địa lý trong mạng cảm biến không dây đã được đề xuất trước đây.
- Tìm hiểu và áp dụng được phương pháp học tăng cường vào trong định tuyến trong mạng cảm biến không dây.
- Tìm hiểu và sử dụng thành thạo các công cụ mô phỏng mạng WiSSim [28], NS2 [29].
- Triển khai hoàn thiện giao thức định tuyến sử dụng học tăng cường trong mạng cảm biến không dây và thu được kết quả mô phỏng tốt.
- Có những kết luận, đánh giá kết quả đạt được của giao thức cũng như những vấn đề cần phải cải thiện trong tương lai.
- Có hai bài báo được đăng trên hội nghị quốc tế:
  - ★ *"Load balanced and constant stretch routing in the vicinity of holes in WSNs"* trong hội nghị CCNC 2018 [18].
  - ★ *"Routing in the Vicinity of Multiple Holes in WSNs"* trong hội nghị ICT-DM 2018 [19] (**Best Student Paper Award**).

## 1.6 Cấu trúc đề án

Ngoài phần Giới thiệu tổng quan (chương 1), đề án được chia thành bốn phần chính như sau:

- Chương 2: Tổng hợp và trình bày về mặt ý tưởng của các giao thức định tuyến tránh hố trong mạng cảm biến không dây đã được đề xuất trước đây.
- Chương 3: Trình bày phương pháp học tăng cường và nếu một số giao thức sử dụng học tăng cường để định tuyến trong mạng cảm biến không dây.
- Chương 4: Trình bày giao thức định tuyến do chúng tôi đề xuất.
- Chương 5: Thí nghiệm và đánh giá hiệu năng của giao thức đề xuất.
- Chương 6: Kết luận.

## Chương 2

# Các giao thức định tuyến tránh hố trong mạng cảm biến không dây

### 2.1 Giao thức xác định hố

Trong một số giao thức định tuyến tránh hố trong mạng cảm biến không dây (GPSR [27], EDGR [20], *etc.*), gói tin có thể định tuyến mà không cần biết trước hình dạng của hố. Các giao thức này tuy có cách xác định đường đi đơn giản nhưng lại không mang lại hiệu quả cao về mặt cân bằng tải cũng như độ dài đường định tuyến. Việc xác định hố trước khi định tuyến giúp các nút mạng có được thông tin cần thiết để xây dựng các đường định tuyến mang lại hiệu quả cao. Giao thức định tuyến mà chúng tôi đề xuất trong đề án này cũng sẽ xác định biên hố trước khi định tuyến.

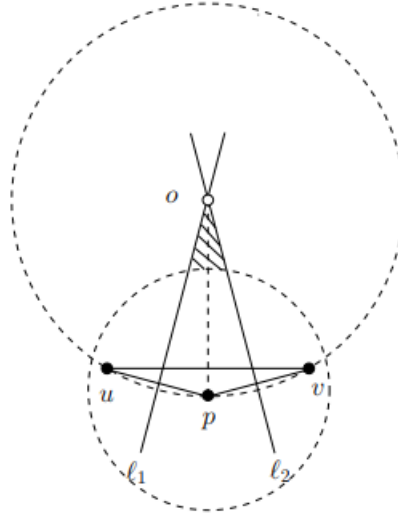
Giao thức xác định hố trong mạng cảm biến không dây đầu tiên là giao thức **BOUND-HOLE** được đề xuất vào năm 2000, bởi Q.Fang *et. al.* [23]. Giao thức này có đầu ra là danh sách các nút theo thứ tự liên tiếp nhau trên biên hố. Ngoài giao thức BOUND-HOLE, một số giao thức xác định hố khác cũng đã được đề xuất như: RollingBall [24], thuật toán xác định biên hố Boundary Critical Point phân tán [25], thuật toán xác định biên hố phân tán Grid [26],... Dựa trên mức độ phổ biến và tính hiệu quả của các giao thức xác định biên hố, trong đề án này chúng tôi lựa chọn sử dụng giao thức BOUNDHOLE. Sau đây sẽ là phần trình bày chi tiết về giao thức.

Giao thức BOUNDHOLE bao gồm 2 giai đoạn: giai đoạn 1 tất cả các nút cảm biến trong mạng sẽ kiểm tra xem nó có khả năng nằm trên biên hố hay không, đồng thời những nút có khả năng nằm trên biên hố cũng sẽ biết được hàng xóm nào có khả năng là nút liền kề nó trên biên hố; giai đoạn 2 những nút biết nó có khả năng nằm trên biên hố sẽ gửi đi gói tin xác định hố (Hole Boundary Detection - HBD) tới các nút liền kề mà nó đã xác định được ở giai đoạn 1. Khi gói tin HBD quay lại nút ban đầu đã tạo ra nó, danh sách tất cả các nút trên biên hố sẽ được xác định.

Ở giai đoạn 1, các nút mạng sẽ kiểm tra mình có nằm trên biên hố không nhờ luật TENT. Tác giả đã đưa ra hai khái niệm về *Strongly Stuck Node* và *luật TENT* như sau:

1. **Strongly Stuck Node** Một nút mạng  $p$  được gọi là *strongly stuck node* nếu tồn

tại một điểm  $q$  nằm ngoài vùng bán kính phủ sóng của  $p$  sao cho không có hàng xóm nào của  $p$  gần  $q$  hơn chính  $p$ . Nút mạng nào là *strongly stuck node* thì sẽ có khả năng nằm trên biên hố.



Hình 2.1: Strongly stuck node

2. **Luật TENT** Luật TENT được dùng để kiểm tra xem một nút mạng có phải là *strongly stuck node* hay không.

Với mỗi nút mạng  $p$ , trước tiên tác giả sắp xếp các hàng xóm của  $p$  theo thứ ngược chiều kim đồng hồ. Và các góc được đề cập ở đây được tính theo ngược chiều kim đồng hồ. Ký hiệu  $B_r(x)$  là đường tròn tâm  $x$  bán kính  $r$ . Gọi  $u, v$  là hai nút hàng xóm của  $p$ . Vẽ 2 đường trung trực của  $up$  và  $vp$ ,  $l_1, l_2$ .  $l_1, l_2$  cắt nhau tại  $o$  và chia mặt phẳng thành 4 phần (hình 4.2). Chỉ những điểm nằm trong góc chứa  $p$  mới gần  $p$  hơn  $u$  và  $v$ . Nếu  $o$  nằm trong vùng bán kính truyền tin của  $p$ , thì với bất kỳ điểm  $q$  nào nằm trong góc  $vpu$  và nằm ngoài bán kính phủ sóng của  $p$ , một trong hai điểm  $u$  hoặc  $v$  sẽ gần  $q$  hơn  $p$ , khi đó  $p$  không phải là *strongly stuck node*. Nói cách khác,  $p$  là *strongly stuck node* khi  $o$  nằm ngoài bán kính phủ sóng của  $p$ . Nhờ luật TENT, những hàng xóm có khả năng nằm liền kề  $p$  trên biên hố ( $u, v$ ) cũng được đánh dấu. Góc  $vpu$  được gọi là *stuck angle*.

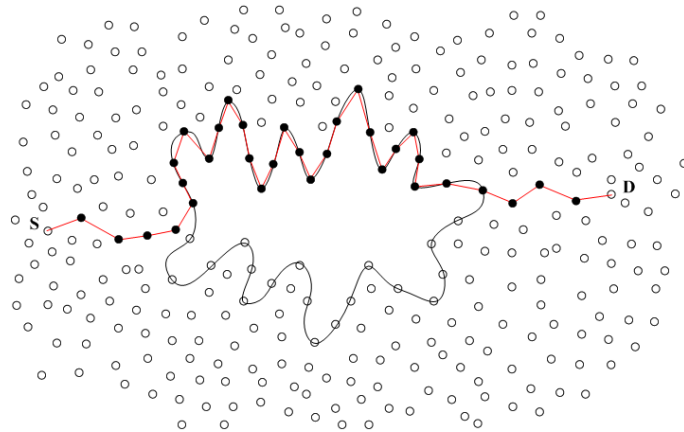
Sau giai đoạn 1, các *strongly stuck node* và các *stuck angle* đã được đánh dấu. Mỗi *strongly stuck node* sẽ tạo một gói tin HBD (Hole Boundary Detection) để xác định biên hố. Gói tin HBD được gửi đi lần lượt tới hàng xóm nằm trong *stuck angle* của *strongly stuck node*, theo ngược chiều kim đồng hồ. Mỗi khi đi tới 1 *strongly stuck node*, gói tin HBD sẽ thêm ID và tọa độ của nút đó vào trong dữ liệu của mình. Khi HBD quay lại nút ban đầu đã tạo ra nó, chúng ta thu được một danh sách các nút trên biên hố theo thứ tự liền kề ngược chiều kim đồng hồ. Để tránh tất cả các *strongly stuck node* trên biên hố đều gửi gói tin HBD đi một vòng xung quanh hố (gây lãng phí năng lượng), mỗi *strongly stuck node* khi nhận được gói tin HBD xuất phát từ nút có tung độ thấp hơn nó, sẽ hủy gói tin HBD đi. Cuối cùng, chỉ có duy nhất một *strongly*

stuck node với tung độ thấp nhất sẽ nhận lại được đầy đủ danh sách các nút trên biên hố.

## 2.2 Giao thức định tuyến tránh hố

### 2.2.1 Tiếp cận theo hướng đi men biên hố

Để tránh hố, các giao thức định tuyến cổ điển tiếp cận theo hướng sử dụng định tuyến tham lam ở những vùng có mật độ nút mạng dày (có thể tìm được hàng xóm gần đích hơn nút hiện tại), và chuyển sang định tuyến men biên hố (perimeter forwarding) khi gặp hố. Khi định tuyến men biên hố, các nút mạng cần xây dựng một đồ thị gọi là *planar graph*, là đồ thị có các cạnh chỉ cắt nhau tại các đầu mút của chúng. Cơ chế định tuyến men biên hố sử dụng planar graph để tìm nút tiếp theo gói tin có thể được chuyển tới mà đường đi không cắt hố. Giao thức GPSR (Greedy Perimeter Stateless Routing) được đề xuất bởi Brad Karp *et al.* vào năm 2000 là giao thức được biến đến nhiều nhất trong tất cả các giao thức tiếp cận theo hướng đi men theo biên hố. Các giao thức tiếp cận theo hướng đi men theo biên hố có thể được tìm thấy ở [1]. Mặc dù các phương pháp tiếp cận này có thể giải quyết được hiện tượng cực tiểu địa phương, nhưng chúng lại gặp phải 2 vấn đề quan trọng. Vấn đề thứ nhất là hiện tượng đường đi bị kéo dài.... Vấn đề thứ hai là đường định tuyến sẽ chỉ tập trung ở biên hố khiến cho cân bằng tải của mạng không tốt. Subarmanian *et al.* [31] đã chỉ ra rằng việc định tuyến men hố này làm giảm đáng kể thông lượng của toàn mạng bởi vì đường truyền tập trung ở biên hố.



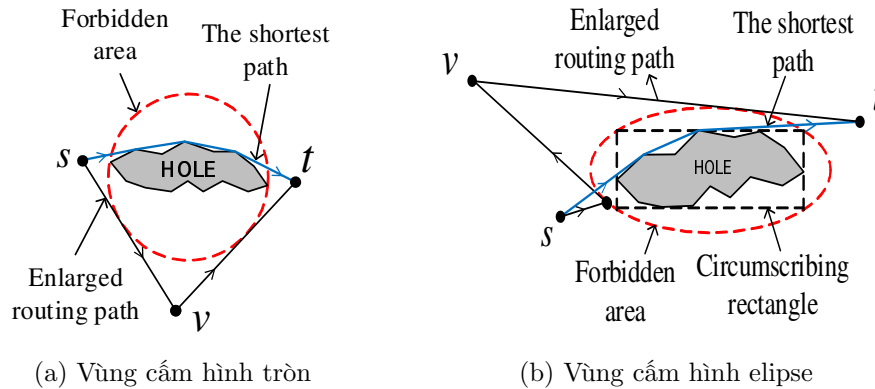
Hình 2.2: Định tuyến men biên hố

### 2.2.2 Tiếp cận theo hướng xây dựng vùng cấm xung quanh hố

Để giải quyết 2 vấn đề của phương pháp định tuyến men biên hố cổ điển, nhiều giao thức định tuyến tiếp cận theo hướng xây dựng vùng cấm xung quanh hố đã được đề xuất như: *BEDH* [33], *HEXAGON* [32], *OCTAGON* [35], etc. Các giao thức này đều dùng một hình hình học đơn giản (hình tròn [33], ellipse [34], lục giác [32], bát giác



[35],...) để bao hồ và làm vùng cấm không cho gói tin đi vào bên trong. Cách làm này nhằm mục đích giảm tải ở biên hồ, và rút ngắn đường định tuyến trong trường hợp hồ có nhiều vùng lõm phức tạp.

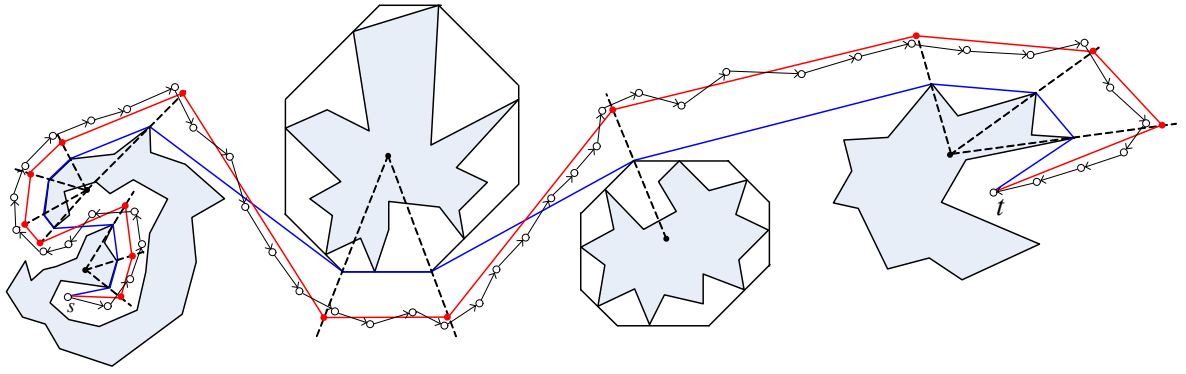


Hình 2.3: Các giao thức định tuyến xây dựng vùng cấm quanh hồ

Tuy vậy, một số giao thức như BEDH [33], ELIPSE [34],... vẫn chưa giải quyết triệt để được vấn đề đường đi bị kéo dài (Hình 2.3). Và hầu hết tất cả các giao thức (BEDH [33], ELIPSE [34], HEXAGON [32],...) cũng chưa giải quyết được triệt để vấn đề tập trung tải ở biên hồ. Bởi thay vì tập trung tải ở biên hồ, các giao thức này lại khiến đường định tuyến tập trung ở vùng biên của vùng cấm.

Khác với các giao thức trên, giao thức OCTAGON [35] không những cho gói tin dữ liệu đi men theo vùng cấm có dạng bát giác, mà còn vị tự bát giác này ra thành những bát giác có kích thước lớn hơn, để gói tin dữ liệu đi men theo những bát giác vị tự ấy. Tỷ lệ vị tự khác nhau cho mỗi lần truyền gói tin dữ liệu sẽ đảm bảo được sự khác nhau của đường định tuyến. Cách làm này đã giải quyết được triệt để 2 vấn đề về độ dài đường định tuyến và cân bằng tải nêu trên. Ngoài ra, gần đây tác giả Phi Lê Nguyễn *et. al.* đã đề xuất giao thức định tuyến đảm bảo cân bằng tải và độ dài đường định tuyến không những cho trường hợp nút nguồn ở ngoài hồ mà còn giải quyết cả các trường hợp nút nguồn ở bên trong hồ [19]. Tác giả đã xác định một tập các đường đi O-clit vượt hồ, vị tự các đường O-clit ấy, rồi chọn ra một đường đi ngẫu nhiên trong số ấy. Đường đi này chứa các đích tạm thời gọi là anchor point, gói tin sẽ đi theo định tuyến tham lam tới các anchor point ấy cho tới khi tới được đích (Hình 2.4). Bằng chứng minh lý thuyết và thực nghiệm, tác giả đã chứng minh được tính hiệu quả về mặt cân bằng tải và tỷ lệ của độ đường đi thực tế so với đường đi O-clit ngắn nhất không vượt quá  $1 + \epsilon$  với  $\epsilon$  là hằng số xác định trước.

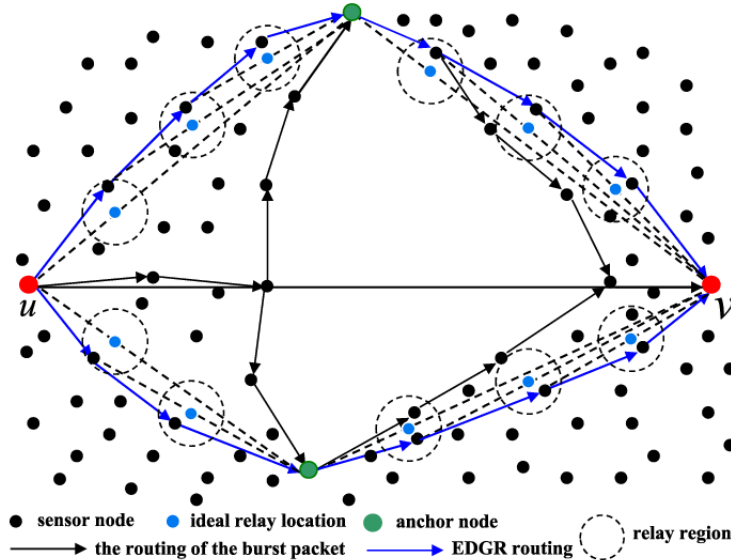
Tuy nhiên, tất cả các giao thức đi theo hướng tiếp cận này đều cần phải phát tán thông tin về hồ, thông tin về đa giác bao phủ của hồ tới các nút mạng nằm xa hồ, để đảm bảo đường đi được xây dựng một cách chính xác. Việc làm này làm tiêu hao năng lượng của các cảm biến.



Hình 2.4: Đường định tuyến của giao thức BSMH. Đường màu xanh da trời là đường O-clit vượt hố cơ bản, đường màu đỏ là đường vị tự của đường cơ bản và cũng chính là đường chứa các anchor point trên thực tế.

### 2.2.3 Tiếp cận theo hướng heuristic

Có khá nhiều giao thức định tuyến sử dụng phương pháp heuristic để đảm bảo gói tin tránh được hố và đảm bảo cân bằng tải cho mạng. Yu *et al.* [21] đã đề xuất một cơ chế để tránh hiện tượng cực tiểu địa phương bằng cách xác định các nút nằm bên trong phần lõm của hố và không cho chúng tham gia vào quá trình định tuyến. Giao thức này chỉ cần ít năng lượng để phát tán thông tin về hố, và cũng tránh được hiện tượng gói tin đi đâm vào hố nhưng nó vẫn vướng phải một vấn đề quan trọng đó là cân bằng tải kém, bởi gói tin chỉ đi men theo phần biên lõm của hố, và đường đi của các gói tin từ cùng một nguồn là luôn giống nhau.



Hình 2.5: Giao thức EDGR [20]

Trong [20], Haojun Huang *et al.* đã đề xuất phương pháp định tuyến trong đó, một vài gói tin "khám phá" sẽ được gửi đi từ nguồn tới đích để xây dựng 2 đường đi cơ bản theo 2 hướng bên trái hoặc bên phải của hố theo hướng từ nguồn tới đích. Hai đường đi cơ bản này tương tự với đường đi O-clit ngắn nhất mà không cắt bao lõm của hố.

Chúng bao gồm các đích tạm thời, sau đây các nút mạng sẽ chọn next hop để truyền gói tin dữ liệu đi lần lượt qua các đích tạm thời này sao cho một điều kiện về năng lượng giữa các nút hàng xóm với nhau được thỏa mãn. Điều kiện về năng lượng này nhằm mục đích cân bằng giữa năng lượng và tần suất gửi tin của các nút mạng, đảm bảo cân bằng tải cho mạng. Tuy vậy, cách giải quyết này lại không đưa ra chứng minh hoặc cơ sở lý thuyết để đảm bảo về độ dài đường định tuyến hay sự đa dạng của đường đi, nhất là trong các trường hợp nút nguồn nằm bên trong vùng lõm của hố. Dựa trên những ưu và nhược điểm của các giao thức định tuyến địa lý đã được phân tích ở trên, trong đề án này chúng tôi sẽ đề xuất một giao thức định tuyến tránh hố - giao thức ***HYBRID***, kết hợp giữa định tuyến địa lý và phương pháp học tăng cường (cụ thể là thuật toán Q-learning), dựa trên các ý tưởng sau:

- Để tiết kiệm năng lượng khi phát tán thông tin hố, chúng tôi sẽ chỉ phát tán thông tin chính xác của hố tới các nút nằm bên trong biên hố. Những nút nằm bên ngoài biên hố sẽ được học thông tin về hố nhờ quá trình back propagation của thuật toán Q-learning.
- Đường định tuyến của các thuật toán trước đây hoặc là sẽ đi men theo biên hố hoặc là sẽ đi dựa trên các đích tạm thời của đường đi được xác định trước. Cách làm thứ nhất không đảm bảo được cân bằng tải cho mạng. Cách làm thứ hai đòi hỏi tính toán nhiều gây tốn thời gian và năng lượng của các nút mạng. Đường định tuyến theo thuật toán HYBRID của chúng tôi sẽ chỉ tính toán ngay tại thời điểm truyền tin dựa trên thông tin của hàng xóm, trong khi vẫn đảm bảo được cân bằng tải của mạng và độ dài của đường đi nhờ vào việc sử dụng Q-learning để cân bằng các yếu tố về năng lượng còn lại, khoảng cách tới đích, khoảng cách tới hố, ... của các nút mạng.

Nguyên lý về phương pháp học tăng cường và cụ thể là thuật toán Q-learning sẽ được trình bày ở ngay phần tiếp theo dưới đây.

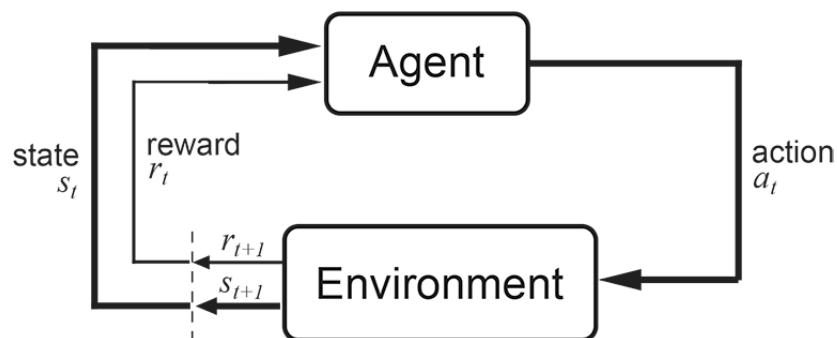
## Chương 3

# Phương pháp học tăng cường

### 3.1 Lý thuyết và ứng dụng của phương pháp học tăng cường

Học tăng cường (*reinforcement learning*) là một trong số các phương pháp học máy, bên cạnh *học có giám sát*, *học bán giám sát* và *học không có giám sát* (phân nhóm dựa trên phương thức học). Học tăng cường là cách thức một đối tượng (*agent*) trong một môi trường (*environment*) chọn cách hành động (*action*) để đạt được hiệu quả tối ưu (hiệu quả tối ưu đối với một mục đích được xác định trước), dựa vào kinh nghiệm hành động từ quá khứ. Phương pháp này là quá trình: "đưa ra hành động - nhận lại phần thưởng từ môi trường - đưa ra hoạt động tối ưu hơn" liên tiếp của đối tượng (Hình 3.1). Phần thưởng (*reward*) là một hàm số có giá trị vô hướng, được tính dựa trên các yếu tố mà đối tượng muốn học. Một cách hình thức, một model học tăng cường bao gồm:

- S: tập các trạng thái (*state*) của môi trường;
- A: tập các hành động (*action*);
- R: tập các khoản thưởng (*reward*) với giá trị vô hướng;



Hình 3.1: Quá trình học tăng cường

Tại mỗi thời điểm  $t$ , đối tượng thấy được trạng thái của nó là  $s_t \in S$  và tập các hành động có thể  $A(s_t)$ . Nó chọn một hành động  $a \in A(s_t)$  và nhận được từ môi trường

trạng thái mới  $s_{t+1}$  và một khoản thưởng  $r_{t+1}$ . Dựa trên các tương tác này, đối tượng học tăng cường phải phát triển một chiến lược  $\pi : S \rightarrow A$  có tác dụng cực đại hóa lượng  $R = r_0 + r_1 + \dots + r_n$  với các quá trình học có trạng thái kết thúc, hoặc lượng  $R = \sum_t \gamma^t r_t$  với các quá trình học không có trạng thái kết thúc (trong đó  $\gamma$  là một hệ số giảm khoản "thưởng trong tương lai" nào đó, với giá trị trong khoảng 0.0 và 1.0). Do đó, học tăng cường thích hợp cho các bài toán có quá trình học lặp lại dài hạn, cần cân bằng được mất giữa các khoản thưởng ngắn hạn và dài hạn. Học tăng cường đã được áp dụng thành công cho nhiều bài toán, trong đó có điều khiển robot, điều vận thang máy, viễn thông, game atari, game mario, cờ vây, cờ vua, ect.

## 3.2 Giới thiệu các thuật toán học tăng cường

Trong cuốn sách *Reinforcement Learning: An Introduction* của Richard S. Sutton và Andrew G. Barto, tác giả chia các phương pháp học tăng cường thành hai nhóm dựa theo không gian trạng thái và hành động của đối tượng:

- **Tabular Solution Methods:** là phương pháp học cho các đối tượng có không gian tập trạng thái và tập hành động đủ nhỏ để các giá trị đánh giá độ tốt của các hành động có thể biểu diễn dưới dạng bảng. Trong nhóm này, các phương pháp có thể tìm ra chính xác giá trị tối ưu nhất mỗi khi thực hiện chiến lược để quyết định hành động tiếp theo.
- **Approximate Solution Methods:** là phương pháp học cho các đối tượng có không gian tập trạng thái và tập hành động lớn, bởi sự giới hạn về khả năng tính toán, agent chỉ có thể tìm được xấp xỉ giá trị tối ưu mỗi khi thực hiện chiến lược để quyết định hành động tiếp theo.

Trong đề án này, chúng tôi lựa chọn sử dụng nhóm phương pháp *Tabular solution methods* bởi lý do: các nút cảm biến chỉ có thể gửi gói tin tới các hàng xóm của nó vì vậy không gian tập hành động của mỗi trạng thái là nhỏ (trong thực tế, mỗi cảm biến trong mạng cảm biến không dây có từ 5  $\rightarrow$  20 hàng xóm).

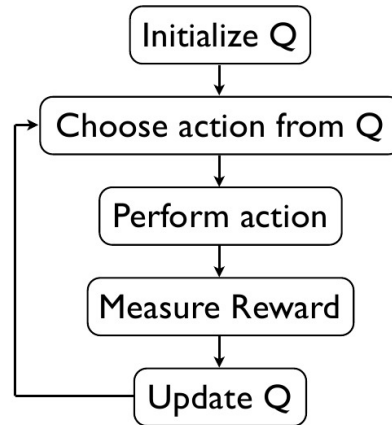
Một số phương pháp học tabular solution method phải kể đến như: Multi-armed Bandits, Finite Markov Decision Processes, Dynamic Programming, Monte Carlo Methods, Temporal-Difference Learning...

## 3.3 Thuật toán Q-learning

**Q-learning** là một trong những thuật toán model-free phổ biến nhất của nhóm thuật toán tabular solution methods. Phương pháp này sử dụng một ma trận  $Q(s, a)$  của các trạng thái và hành động, mỗi một cặp (*trạng thái, hành động*) sẽ có một giá trị gọi là Q-value thể hiện độ tốt của việc thực hiện hành động  $a$  khi đối tượng đang ở trạng thái  $s$ . Giá trị Q-value được cập nhật bởi công thức sau:

$$Q(s, a) \leftarrow (1 - \alpha).Q(s, a) + \alpha.[r + \gamma.\max_{a'} Q(s', a')] \quad (3.1)$$

Trong đó:  $s'$  là trạng ngay tiếp theo của trạng thái  $s$ ,  $\alpha$  là **learning rate**,  $r$  là **reward** và  $\gamma$  là **discount factor**,  $\max_{a'} Q(s', a')$  là giá trị Q-value lớn nhất khi đối tượng đang ở trạng thái  $s'$ , được gọi là V-value của  $s'$ . Quá trình học của thuật toán Q-learning được thể hiện ở sơ đồ dưới đây.



Hình 3.2: Quá trình học của phương pháp Q-learning

### 3.4 Các giao thức định tuyến sử dụng học tăng cường trong mạng cảm biến không dây

Một số giao thức định tuyến trong mạng cảm biến không dây sử dụng học tăng cường đã được đề xuất, hầu hết các tác giả đều sử dụng phương pháp Q-learning, như: *Adaptive Routing for Sensor Networks using Reinforcement Learning* (2006), *Reinforcement Learning Based Geographic Routing Protocol for UWB Wireless Sensor Network* (2007), *QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks* (2010). Trong các giao thức này, tác giả định nghĩa tập các trạng thái, hành động, môi trường của mô hình học tăng cường như sau:

- **agent**: là các gói tin dữ liệu
- **enviroment**: là tập hợp tất cả các nút mạng
- **state  $s$** : là trạng thái gói tin đi tới một nút  $s$  trong mạng
- **action  $a$** : là sự kiện nút  $a$  được chọn làm next hop để gửi gói tin

Sau mỗi lần gói tin được gửi đi, Q-value của hàng xóm mà nó vừa gửi gói tin tới sẽ được cập nhật theo công thức 3.1. Công thức 3.1 là công thức chung cho tất cả các thuật toán Q-learning, yếu tố quyết định tính hiệu quả giữa các thuật toán chính là cách xây dựng hàm tính reward. Trong các thuật toán định tuyến sử dụng Q-learning đã được đề xuất, tác giả xây dựng hàm tính reward dựa trên 2 yếu tố: năng lượng còn lại và khoảng cách tới đích của các hàng xóm, và không giải quyết trường hợp mạng

có hố. Tuy giao thức được đề xuất trong [36] có đề cập tới việc giải quyết hiện tượng cực tiểu địa phương dựa vào việc drop gói tin và gửi lại một reward rất thấp mỗi khi gói tin gặp stuck node, cách làm này chỉ hiệu quả khi mạng chỉ tồn tại một vài vùng có mật độ sensor thấp khiến cho gói tin đi vào đó gặp phải hiện tượng cực tiểu địa phương, còn trong trường hợp mạng có hố rộng và phức tạp, giao thức này phải drop rất nhiều gói tin trước khi tìm được đường định tuyến tránh hố hợp lý, dẫn đến tỷ lệ truyền tin thành công thấp.

Sau đây, chúng tôi sẽ đề xuất một giao thức định tuyến trong mạng cảm biến không dây có hố: *giao thức **HYBRID***. Giao thức này là sự kết hợp giữa định tuyến địa lý và phương pháp Q-learning. Bằng thực nghiệm, chúng tôi sẽ chứng minh rằng giao thức đề xuất đảm bảo được 3 yếu tố sau:

- *Tỷ lệ truyền tin thành công cao (trên 99%).*
- *Thời gian truyền tin từ nguồn tới đích ngắn.*
- *Thời gian sống của mạng được kéo dài hơn so với các giao thức trước đây.*

## Chương 4

# Đề xuất giao thức định tuyến tránh hố trong mạng cảm biến không dây sử dụng phương pháp học tăng cường

Trong chương này, chúng tôi sẽ trình bày giao thức định tuyến sử dụng phương pháp học tăng cường để tìm đường đi trong các mạng cảm biến không dây có sự tồn tại của hố. Như đã đề cập ở các phần trước, mục tiêu xây dựng giao thức của chúng tôi là kéo dài thời gian sống cho mạng. Thời gian sống của mạng được tính từ thời điểm mạng bắt đầu hoạt động cho tới khi nút đầu tiên chết. Giao thức được đề xuất dưới đây sẽ đảm bảo các yếu tố sau:

- Đảm bảo cân bằng tải cho mạng để tránh tình trạng một nút phải làm việc quá tải dẫn đến hết năng lượng nhanh chóng.
- Đảm bảo đường đi không quá dài để tránh tốn kém năng lượng cho toàn mạng.
- Tính toán đơn giản để giảm thời gian truyền tin từ nguồn tới đích, đồng thời cũng tiết kiệm năng lượng cho các nút mạng.

Giả thiết bài toán sẽ được trình bày ở chương 3.1, bao gồm hai phần: mô tả mô hình mạng và giới thiệu các ký hiệu - định nghĩa. Tiếp theo, ở chương 3.2 chúng tôi sẽ trình bày tổng quan về giao thức. Và tiếp theo đó chính là phần mô tả chi tiết về các pha của giao thức bao gồm: chương 3.1 mô tả pha phát hiện hố, chương 3.2 mô tả pha phát tán thông tin về vùng lõm của hố, chương 3.3 mô tả phần trọng tâm của giao thức: pha truyền tin (truyền dữ liệu), cuối cùng chương 3.4 mô tả phần gửi nhận gói tin điều khiển để các node cập nhật thông tin về hàng xóm của nó.

### 4.1 Giả thiết bài toán

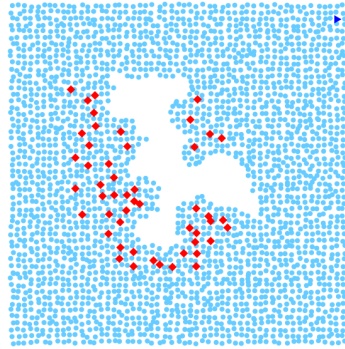
Chúng tôi giả sử rằng ban đầu tất cả các nút mạng đều không biết trước thông tin về hố.



### 4.1.1 Mô hình mạng

Mô hình mạng được xét tới trong luận văn này là các mạng có sự xuất hiện của hố (hình 4.1), có nhiều nút nguồn để thu thập thông tin về mục tiêu, và có 1 nút đích duy nhất (một base station) để xử lý dữ liệu đã thu thập được từ các nút nguồn.

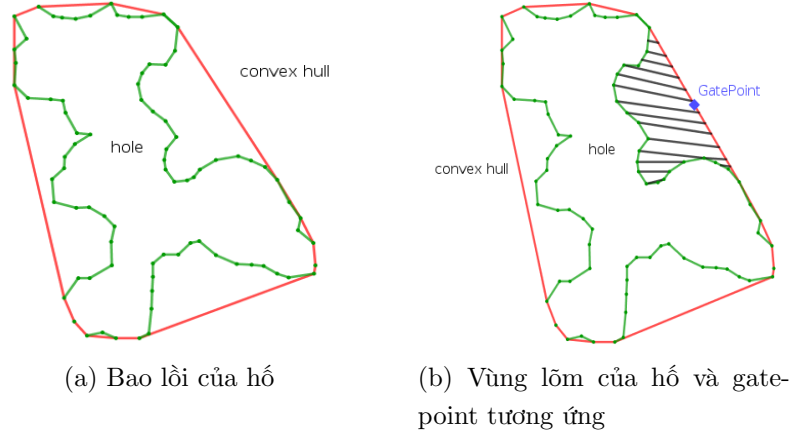
Giả sử rằng mỗi node đều biết vị trí của nó (sử dụng GPS hoặc các dịch vụ định vị khác [22]) và vị trí hàng xóm của nó (nhờ quá trình trao đổi gói tin về tọa độ giữa các hàng xóm với nhau). Ngoài ra, nút nguồn biết được vị trí của nút đích.



Hình 4.1: Mô hình mạng có sự xuất hiện của hố, các nút hình thoi màu đỏ là nguồn, nút màu xanh nước biển góc trên cùng bên phải của mạng là đích.

### 4.1.2 Các ký hiệu và định nghĩa

- **Khoảng cách Ó-clit giữa hai điểm:** Khoảng cách Ó-clit giữa 2 điểm  $A, B$  chính là độ dài của đoạn thẳng  $AB$ .
- **Hố định tuyến (hố):** hố định tuyến là một đa giác có đỉnh là các nút cảm biến  $S_1, S_2, \dots, S_n$ , có các cạnh không cắt nhau tại điểm khác đầu mút của chúng, và thỏa mãn điều kiện:
  - Không có nút cảm biến nào nằm bên trong đa giác này
  - Khoảng cách Ó-clit giữa  $S_i$  và  $S_{i+1}$  không vượt quá bán kính truyền tin của các nút cảm biến ( $\forall i = \overline{1, n}; S_{n+1} \equiv S_1$ )
- **Bao lồi của hố:** là đa giác lồi có diện tích nhỏ nhất thỏa mãn điều kiện hố nằm hoàn toàn trong đa giác này và có đỉnh là các đỉnh của hố.



Hình 4.2: Các định nghĩa liên quan tới hố

- **Vùng lõm của hố** : là tập hợp các nút liên tiếp trên biên hố nằm giữa 2 đỉnh liên tiếp trên bao lồi của hố (Hình 4.1 (b))(Lưu ý: chúng tôi không xét những vùng lõm có số đỉnh nhỏ hơn 15).
- **Potential stuck node**: *potential stuck node* là những điểm nằm trong vùng lõm của hố. Trong trường hợp gói tin dữ liệu xuất phát từ nút nằm ngoài vùng lõm của hố, các potential stuck node sẽ không tham gia vào quá trình truyền tin.
- **Gate-point của vùng lõm**: là trung điểm của đoạn thẳng nối điểm bắt đầu và kết thúc của vùng lõm (Hình 4.1 (b)).
- **Khoảng cách ngắn nhất tới gate-point**: khoảng cách ngắn nhất từ một điểm nằm bên trong vùng lõm  $\mathcal{R}$  tới gate-point của  $\mathcal{R}$  là độ dài đường đi ngắn nhất giữa 2 điểm này mà không cắt  $\mathcal{R}$  (được xác định theo thuật toán tìm đường đi ngắn nhất giữa 2 điểm trong đa giác [30]).
- **Next hop**: *Next hop* là hàng xóm mà nút hiện tại chọn ra để truyền gói tin tới.
- **Stuck node**: nút mạng mà gói tin đi đến đó và không tìm được next hop.

## 4.2 Tổng quan

Giả sử rằng ban đầu tất cả các nút mạng đều không có thông tin về hố. Như đã nêu ở chương 2.1, khi gói tin dữ liệu đâm vào trong vùng lõm của hố và gặp *stuck node*, nó sẽ phải đi vòng ra khiến đường đi bị kéo dài đáng kể. Trong giao thức HYBRID này, để gói tin dữ liệu có thể tránh hố hiệu quả, trước pha truyền tin dữ liệu, chúng tôi thiết kế 2 pha setup là: *Xác định hố* và *Phát tán thông vùng lõm của hố*. Cụ thể, giao thức HYBRID của chúng tôi bao gồm 3 giao thức nhỏ theo thứ tự lần lượt như sau:

- **Giao thức xác định hố**: Thu thập thông tin về tất cả các nút trên biên hố. Mục đích của giao thức xác định hố là để phục vụ cho việc xác định các vùng lõm của hố ở pha sau.

- **Giao thức phát tán thông tin vùng lõm của hố:** Trong giao thức này, danh sách các vùng lõm của hố sẽ được xác định và thông tin của mỗi vùng lõm sẽ được phát tán cho những nút nằm bên trong vùng lõm đó. Qua việc phát tán thông tin này, những nút nằm trong vùng lõm của hố sẽ đánh dấu chính nó là *potential stuck node* và tạo thành một vùng cấm ngăn không cho gói tin dữ liệu đi từ ngoài vào bên trong. Bên cạnh đó, những gói tin dữ liệu xuất phát từ bên trong vùng lõm của hố cũng có thể dựa vào thông tin về vùng lõm đã nhận được để thoát hố một cách hiệu quả.
- **Giao thức truyền tin:** Ở trong giao thức truyền tin dữ liệu, các nút mạng sẽ sử dụng thông tin về hố đã được xác định ở 2 giao thức đầu để tránh hố. Đồng thời, chúng cũng sử dụng phương pháp Q-learning để học cách tối ưu đường đi, bao gồm: rút ngắn độ dài đường đi, tránh các nút biên hố, chọn các nút mạng có năng lượng còn lại cao, chọn các nút mạng có tần suất nhận tin thấp để làm next hop.

### 4.3 Giao thức xác định hố

Pha phát hiện biên hố được thực hiện theo thuật toán *BOUNDHOLE* [23], như đã trình bày ở chương 2.1.1. Luật *TENT* được sử dụng để các nút mạng kiểm tra xem mình có khả năng nằm trên biên hố hay không. Những nút mạng có khả năng nằm trên biên hố sẽ tạo gói tin *Hole Boundary Detection (HBD)* và truyền gói tin ấy đi vòng quanh biên hố để thu thập thông tin về các nút nằm trên biên hố. Khi gói tin HBD quay trở lại nút đã tạo ra nó, ta thu được một danh sách đầy đủ các nút trên biên hố. Để tránh việc tất cả các gói HBD đều đi hết một vòng quanh hố, các nút mạng sẽ hủy những gói tin xuất phát từ một nút có tung độ thấp hơn nó. Cuối cùng chỉ có duy nhất một nút có tung độ cao nhất thu được thông tin đầy đủ về hố. Ta gọi nút này là *leader node*  $H_0$ .

### 4.4 Giao thức phát tán thông tin vùng lõm của hố

Mục tiêu của giao thức phát tán thông tin hố là làm cho các nút nằm ở trong vùng lõm của hố biết được nó là *potential stuck node* và những hàng xóm nào của nó là *potential stuck node*. Cách làm của chúng tôi đó là sau khi danh sách các nút trên biên hố đã được xác định, danh sách này được gửi đi 1 vòng lần lượt qua các nút trên biên hố, khi gặp 1 nút là điểm bắt đầu của vùng lõm, nút đó sẽ xác định vùng lõm bắt đầu từ nó, sau đó phát tán thông tin về vùng lõm cho tất cả các nút nằm bên trong vùng lõm đó. Nút nào nhận được thông tin phát tán này sẽ đánh dấu chính nó và hàng xóm của nó là *potential stuck node*.

Giao thức phát tán thông tin vùng lõm của hố bao gồm các bước sau:

- **Bước 1 - Gửi gói tin Refresh tới các điểm bắt đầu của vùng lõm:** *leader node*  $H_0$  xác định các nút bắt đầu của các vùng lõm, sau đó tạo 1 gói tin

REFRESH, chèn danh sách các nút trên biên hố và danh sách các nút bắt đầu vùng lõm vào trong gói tin REFRESH. Rồi gửi gói tin REFRESH này đi lần lượt qua từng nút trên biên hố.

- **Bước 2 - Nút bắt đầu vùng lõm xác định vùng lõm của nó:** Khi 1 nút nhận được gói tin REFRESH, nó sẽ lấy ra thông tin danh sách các nút bắt đầu vùng lõm từ trong gói tin này, kiểm tra xem mình có nằm trong danh sách đó không. Nếu có, nó sẽ xác định vùng lõm chứa nó; sau đó tạo 1 gói tin BROADCAST, chèn danh sách các nút nằm trên vùng lõm vào gói tin BROADCAST này và phát tán đi.
- **Bước 3 - Phát tán thông tin vùng lõm cho các nút nằm bên trong nó:** Khi 1 nút nhận được gói tin BROADCAST, nó sẽ lấy danh sách các nút trong vùng lõm ra, và kiểm tra xem mình có nằm bên trong vùng lõm này không, nếu có nó tiếp tục phát tán gói tin BROADCAST.
- **Bước 4 - Lưu các thông tin cần thiết cho pha định tuyến:** Khi 1 nút nhận được gói tin BROADCAST, ngoài việc xác định xem có phát tán gói tin đi tiếp hay không, nó còn cần phải lưu các thông tin cần thiết để phục vụ cho pha định tuyến. Cụ thể nó sẽ làm các công việc sau:
  - Đánh dấu chính nó và hàng xóm gửi gói tin BROADCAST tới nó là *potential stuck node*
  - Xác định đường đi ngắn nhất từ nó và từ hàng xóm tới *gate-point* của vùng lõm, và lưu lại.

---

**Algorithm 1:** Xác định vùng lõm của hố

---

**Data:** *hole* - list of hole's boundary nodes

**Result:** *ID of next hop*

```

1 cavernList = [ ];
2 for i in hole do
3   cavern = [ ];
4   if i.isConvexHullBoundary & !(i + 1).isConvexHullBoundary then
5     cavern.append(i++) while !i.isConvexHullBoundary do
6       cavern.append(i);
7     end
8   end
9   if cavern.size() > MIN_CAVE_VERTEX then
10    cavernList.append(cavern)
11  end
12 end
13 return cavernList

```

---

Ở trong pha này chúng tôi sử dụng 2 phương thức: *phương thức xác định vùng lõm của hố* (ở Bước 1, Bước 2) được trình bày ở **Algorithm 1** và *phương thức tìm đường đi ngắn nhất trong đa giác* (Bước 4) được giới thiệu trong [30].

## 4.5 Giao thức truyền tin

Giao thức truyền tin là phần chính trong giao thức do chúng tôi đề xuất. Giao thức này có nhiệm vụ truyền các gói tin dữ liệu từ nguồn tới đích.

Như đã trình bày ở chương 1.4 *Phương hướng và mục tiêu nghiên cứu của đồ án*, giao thức định tuyến mà chúng tôi đề xuất là sự kết hợp giữa định tuyến địa lý và phương pháp học tăng cường, cụ thể là sử dụng thuật toán Q-learning. Trong đó, định tuyến địa lý làm nhiệm vụ đảm bảo được các yếu tố: đường đi của gói tin không bị lặp lại và luôn hướng về đích, gói tin thoát và tránh được hố; và Q-learning sẽ làm nhiệm vụ tối ưu hóa đường định tuyến và đảm bảo cân bằng tải cho các nút mạng (ví dụ như chọn đường đi ngắn, tạo ra sự đa dạng giữa đường định tuyến của các gói tin xuất phát từ cùng một nút nguồn để phân bố tải đều trên toàn mạng...).

Dựa theo phương pháp Q-learning, đối với mỗi nút mạng, chúng tôi xây dựng một bảng Q-value chứa các Q-value của hàng xóm của nó. Q-value thể hiện độ tốt của hàng xóm đối với đích. Bảng Q-value được cập nhật sau mỗi lần nút mạng gửi gói tin đi. Khi một nút cần truyền gói tin đi, ban đầu nó sẽ sử dụng định tuyến địa lý để xác định ra tập các hàng xóm mà nó có thể truyền gói tin tới, sau đó chọn trong tập các hàng xóm này ra một nút có Q-value lớn nhất để làm next hop. Cụ thể về các cơ chế truyền tin và việc sử dụng Q-learning để tối ưu hóa đường định tuyến sẽ được trình bày ở chương 4.5.1 và chương 4.5.2 tương ứng dưới đây.

### 4.5.1 Các cơ chế truyền tin

Để phục vụ cho việc định tuyến thoát hố và chuyển hướng khi gặp hố, chúng tôi đã thiết kế 3 cơ chế định tuyến: **TOWARD DESTINATION** (hướng tới đích) được sử dụng bất cứ khi nào có thể; **BYPASS HOLE** (tránh hố) được sử dụng khi không tìm được next hop theo cơ chế *toward destination*; **ESCAPE HOLE** (thoát hố) được sử dụng khi truyền tin ở bên trong vùng lõm của hố.

Khi một nút muốn tìm next hop để gửi gói tin tới, trước tiên nó sẽ kiểm tra xem có hàng xóm nào của nó là *potential stuck node* hay không. Có hai trường hợp xảy ra:

- Nếu tất cả các hàng xóm đều là *potential stuck node*, nghĩa là nút hiện tại nằm bên trong vùng lõm của hố, nó sẽ tìm next hop theo cơ chế Escape Hole để thoát hố.
- Nếu tồn tại hàng xóm không phải *potential stuck node*, nghĩa là nút hiện tại không nằm bên trong vùng lõm của hố, trước tiên nó sẽ tìm next hop theo cơ chế Toward Destination. Khi không tìm được next hop theo cơ chế Toward Destination, nút mạng sẽ chuyển sang tìm next hop theo cơ chế Bypass Hole.

---

**Algorithm 2:** Tìm next hop

---

**Result:** *ID of next hop*

```
14 int nextHop;  
15 bool k = checkAllNeighborsIsPotentialStuckNode();  
16 if k then  
17   | nextHop  $\leftarrow$  findNextHopByHoleEscape();  
18 end  
19 else  
20   | nextHop  $\leftarrow$  findNextHopByTowardDestination();  
21   | if nextHop == NULL then  
22     | nextHop  $\leftarrow$  findNextHopByBypassHole();  
23   | end  
24 end  
25 return nexthop.id
```

---

Ý tưởng chung của cả ba cơ chế trên đều là lập ra một danh sách các hàng xóm thỏa mãn điều kiện ràng buộc về vị trí địa lý đối với hố và đối với nút đích, sau đó chọn ra từ danh sách này một nút có Q-value lớn nhất để làm next hop.

Chi tiết về các cơ chế sẽ được trình bày ở dưới đây.

### (i) Cơ chế Toward Destination

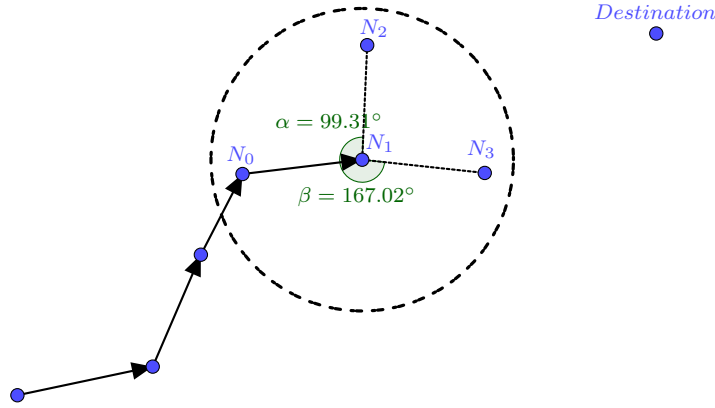
Trong cơ chế truyền tin *Toward Destination*, nút mạng sẽ chọn từ các hàng xóm của nó một danh sách  $L$  các hàng xóm thỏa mãn đồng thời ba điều kiện sau:

1. Không phải *potential stuck node*.
2. Có khoảng cách O-clit tới đích gần hơn khoảng O-clit từ nút hiện tại tới đích.
3. Ký hiệu nút hiện tại là  $N_0$ , nút ngay liền trước nút hiện tại trong đường định tuyến là  $N_1$ , và hàng xóm đang xét là  $N_i$ . Thì hàng xóm  $N_i$  này phải thỏa mãn điều kiện góc  $\widehat{N_0N_1N_i} > threshold$  ( góc  $\widehat{N_0N_1N_i}$  được tính theo góc tuyệt đối, tức là lấy góc nhọn). Trong đó *threshold* là hằng số lớn hơn  $90^\circ$  được xác định trước.

Sau đó, nút mạng sẽ chọn từ trong danh sách  $L$  một hàng xóm có Q-value lớn nhất để truyền gói tin tới (thuật toán 3).

Điều kiện (1) đảm bảo gói tin không đi vào bên trong vùng lõm của hố, điều kiện (2) đảm bảo gói tin luôn tiến tới gần đích chứ không đi ra xa, và điều kiện (3) nhằm mục đích giảm độ dài đường đi cho gói tin. Ở trong ví dụ hình 4.3, đường mũi tên màu đen biểu thị đường đi của gói tin,  $N_1$  là nút hiện tại,  $N_0$  là nút ngay trước nút hiện tại, nét đứt hình tròn thể hiện bán kính truyền tin của  $N_1$ ;  $N_2, N_3$  là các hàng xóm của  $N_1$ . Giả sử  $threshold = 120^\circ$ .  $N_2$  và  $N_3$  đều thỏa mãn điều kiện 1 và 2 (giả sử các nút mạng trong hình đều ở xa hố), nhưng  $N_2$  tạo với  $N_0$  và  $N_1$  góc  $\alpha = 99.31^\circ < 120^\circ$ , không thỏa mãn điều kiện (c), trong khi đó  $N_3$  tạo với  $N_0$  và  $N_2$  góc  $\beta = 167.02^\circ > 120^\circ$ ,

thỏa mãn điều kiện (c). Vậy chỉ có  $N_3$  được chọn vào trong danh sách L. Nếu không có điều kiện (c), trường hợp sau có thể xảy ra: nút  $N_1$  chọn  $N_2$  để làm next hop, sau đó  $N_2$  chọn  $N_3$  làm next hop, như vậy đường đi của gói tin sẽ bị kéo dài.



Hình 4.3: Minh họa cách chọn next hop của cơ chế toward destination

---

**Algorithm 3:** Cơ chế Toward Destination

---

**Data:** *dest*: coordinate of destination  
*prev*: coordinate of previous node on routing path  
*this\_node*: coordinate of this node

**Result:** *ID of next hop*

```

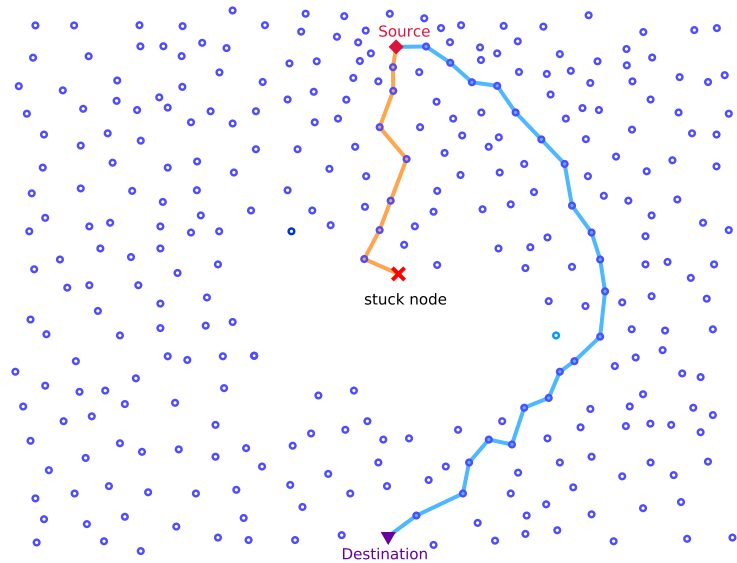
26 candidate_nexthops = []
27 for n in neighbor_list do
28     if n.is_not_potential_stuck_node & distance(n, dest) < distance(this_node,
        dest) & absolute_angle(prev, this_node, n) < threshold then
29         candidate_nexthops.append(n)
30     end
31 end
32 if candidate_nexthops is empty then
33     return -1
34 end
35 nexthop = choose node with max q-value from candidate_nexthops
36 return nexthop.id

```

---

So sánh với cơ chế Greedy cổ điển, thay vì chọn next hop là hàng xóm có khoảng cách tới đích nhỏ nhất, cơ chế Toward Destination của chúng tôi chọn next hop là hàng xóm có Q-value lớn nhất. Trong đó Q-value là giá trị đã được học dựa trên các yếu tố khoảng cách tới đích, khoảng cách tới hồ, hàng xóm có khả năng là stuck node hay không,... Vì vậy, sau một vài gói tin dữ liệu ban đầu được truyền đi, cơ chế Toward

Destination của chúng tôi đã có thể học được cách tránh hố hay từ đầu chứ không đi đâm vào hố như cơ chế Greedy(hình 4.4).



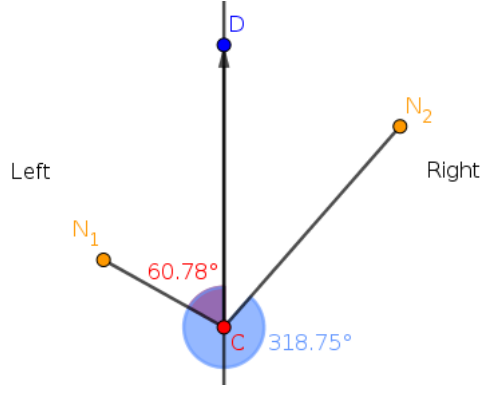
Hình 4.4: So sánh giữa đường đi greedy cổ điển và đường đi toward destination của HYBRID. Đường đi greedy (màu cam) luôn chọn nút gần đích nhất trong các hàng xóm nên sẽ có khả năng đi đâm vào hố và gặp stuck node. Đường đi toward destination của chúng tôi (màu xanh da trời) nhờ chọn nút có Q-value lớn nhất, thì sau một thời gian, nút mạng đã có thể học được cách định tuyến tránh hố ngay từ đầu.

## (ii) Cơ chế BYPASS HOLE

Khi nút mạng không tìm được next hop theo cơ chế *toward destination*, nó sẽ chuyển sang tìm next hop theo cơ chế *bypass hole*. Việc này thường xảy ra khi gói tin gặp hố. Cơ chế Toward Destination luôn tìm hàng xóm gần đích hơn nó để chuyển gói tin tới, nên các nút mạng nằm trên đường định tuyến luôn gần đích hơn các nút trước đó, vì vậy gói tin sẽ không bao giờ đi vòng lại. Tuy nhiên, khi đi theo cơ chế Bypass Hole, để tránh hố, gói tin có thể phải chuyển tiếp gói tin tới nút xa đích hơn nó. Để tránh trường hợp gói tin đi vòng lại khi đi theo cơ chế Bypass Hole, chúng tôi đặt thông tin hướng đi tới đích của gói tin dữ liệu vào header của nó. Có hai hướng đi tới đích của gói tin là *Trái* và *Phải* được định nghĩa như sau (xem Hình 4.4):

- Ký hiệu nút hiện tại đang có gói tin dữ liệu là C, đích là D. Nếu next hop nằm ở nửa mặt phẳng bên trái của tia CD, gói tin sẽ có hướng đi là bên trái. Nếu next hop nằm ở nửa mặt phẳng bên phải của tia CD, gói tin sẽ có hướng đi là bên phải. Gói tin sẽ luôn giữ hướng đi đã được xác định từ ban đầu cho tới khi quay lại được cơ chế Toward Destination.
- Về tính toán hình học, những nút nằm ở nửa mặt phẳng bên trái của đường thẳng CD sẽ có tính chất: lấy tâm là C, góc quay ngược chiều kim đồng hồ từ D tới nút ấy nhỏ hơn  $180^\circ$ ; và những nút nằm ở nửa mặt phẳng bên phải của đường thẳng CD có tính chất ngược lại. Ví dụ trong Hình 4.2,  $N_1$  nằm ở bên trái đường thẳng CD có góc  $(D, C, N_1) = 60.78^\circ < 180^\circ$ ,  $N_2$  nằm ở bên phải đường thẳng CD có góc  $(D, C, N_2) = 318.75^\circ > 180^\circ$ .





Hình 4.5: Định nghĩa hướng đi bên phải và bên trái của gói tin

Bởi vì chỉ khi không đi được theo cơ chế Toward Destination, gói tin mới đi theo cơ chế Bypass Hole, nên trong cơ chế này, nút mạng chỉ chọn từ trong số các hàng xóm xa đích hơn nó để truyền tin tới. Cụ thể, nút mạng sẽ lập ra một danh sách  $L$  các hàng xóm của nó thỏa mãn điều kiện:

1. Không phải *potential stuck node*.
2. Có khoảng cách tới đích xa hơn khoảng cách từ nút hiện tại tới đích.
3. Thuộc nửa mặt phẳng bên trái của tia CD (ký hiệu nút hiện tại là C, nút đích là D) nếu hướng đi của gói tin là bên trái, và thuộc nửa mặt phẳng bên phải của tia CD nếu hướng đi của gói tin là bên phải.

Sau đó, nút mạng sẽ chọn từ trong danh sách  $L$  một hàng xóm có Q-value lớn nhất để truyền gói tin tới.

---

**Algorithm 4:** Find Direction To Dest

---

**Data:** *dest*: coordinate of destination;  $L_1$ : list of candidate next hops;

**Result:** *direction to destination*

```

37 neighbor = neighborWithShortestDistanceToDestination( $L_1$ , dest)
38 if isLeftDirection(neighbor, destination) then
39   | direction = Left
40 end
41 else
42   | direction = Right
43 end
44 return direction

```

---

---

**Algorithm 5:** Cơ chế Bypass Hole

---

**Data:** *dest*: coordinate of destination; *direction*: direction of packet toward destination; *this\_node*: coordinate of this node

**Result:** *ID of next hop*

```
45  $L_1 = []$ ; candidate_nexthops = [];  
46 for  $n$  in neighbor_list do  
47     if  $n.is\_not\_potential\_stuck\_node \ \&\& \ distance(n, dest) \geq$   
          $distance(this\_node, dest)$  then  
48          $L_1.append(n)$   
49     end  
50 end  
51 // first time switch to Bypass Hole mode  
52 if  $direction == None$  then  
53      $direction = findDirectionToDest(L_1, dest)$   
54 end  
55 switch direction do  
56     case Left do  
57         for  $n$  in  $L_1$  do  
58             if  $angle(this\_node, dest, n) < 180^\circ$  then  
59                 candidate_nexthops.append( $n$ )  
60             end  
61         end  
62     end  
63     case Right do  
64         for  $n$  in  $L_1$  do  
65             if  $angle(this\_node, dest, n) \geq 180^\circ$  then  
66                 candidate_nexthops.append( $n$ )  
67             end  
68         end  
69     end  
70     otherwise do  
71         return-1  
72     end  
73 end
```

---

**(iii) Cơ chế ESCAPE HOLE**

Cơ chế *escape hole* được đặc biệt thiết kế cho các trường hợp nút nguồn nằm bên trong vùng lõm của hố. Khi định tuyến theo cơ chế Escape Hole, một nút sẽ xác định một danh sách  $L$  các hàng xóm của nó thỏa mãn điều kiện: *khoảng cách ngắn nhất từ hàng xóm tới gate-point nhỏ hơn khoảng cách ngắn nhất từ nút hiện tại tới gate-point*. Sau đó, từ tập  $L$ , nút có  $Q$ -value cao nhất sẽ được chọn làm nút tiếp theo để gửi gói tin dữ liệu tới.

---

**Algorithm 6:** Cơ chế Escape Hole

---

**Result:** *ID of next hop*

```
74 candidate_nexthops = []
75 for n in neighbor_list do
76     if n.shortest_way_to_gate_point < self.shortest_way_to_gate_point then
77         append n to candidate_nexthops
78     end
79 end
80 if candidate_nexthops is empty then
81     return -1
82 end
83 nexthop = choose node with max q-value from candidate_nexthops
84 return nexthop.id
```

---

#### 4.5.2 Sử dụng Q-learning để tối ưu hóa đường định tuyến của gói tin

Trong chương này chúng tôi sẽ trình bày cách áp dụng phương pháp Q-learning vào định tuyến trong mạng cảm biến không dây.

Chúng tôi định nghĩa *agent*, *environment*, *state* ***s*** và *action* ***a*** như sau:

- ***agent***: là các gói tin dữ liệu
- ***enviroment***: là tập hợp tất cả các nút mạng
- ***state s***: là trạng thái gói tin đi tới một nút ***s*** trong mạng
- ***action a***: là sự kiện nút ***a*** được chọn làm next hop để gửi gói tin

Trong việc tối ưu hóa đường định tuyến và đảm bảo cân bằng tải cho các nút mạng, chúng tôi quan tâm tới 4 yếu tố sau:

(i) Khoảng cách O-clit tới đích của các nút mạng trên đường định tuyến

Khoảng cách tới đích của các nút mạng càng gần, độ dài đường đi tới đích càng giảm.

(ii) Khoảng cách tới hố của các nút mạng trên đường định tuyến

Khi gặp hố, gói tin có xu hướng không đi được theo cơ chế *Toward Destination* nữa và phải chuyển sang cơ chế *Bypass Hole* để đi vòng qua biên hố. Điều này vừa kéo dài đường định tuyến vừa làm giảm năng lượng của các nút trên biên hố dẫn đến hiện tượng hố bị lan rộng. Vì vậy, trong cơ chế định tuyến của chúng tôi, những nút mạng nằm ở xa hố sẽ được ưu tiên chọn làm next hop để gửi gói tin tới.

Chúng tôi ký hiệu chỉ số  $h_{index}$  để thể hiện khoảng cách tới hố của các nút mạng. Mỗi nút mạng cần lưu  $h_{index}$  của chính nó và của các hàng xóm của nó. Khi khởi tạo mạng, tất cả các nút mạng sẽ gán  $h_{index}$  của mình bằng  $MAX\_INTEGER$ . Sau đó, đến pha

phát tán thông tin vùng lân của hồ (chương 4.3), những nút nhận được gói tin Refresh sẽ biết mình nằm trên biên hồ, và gán lại  $h_{index} = 0$ . Những nút mạng không nằm trên biên hồ sẽ cập nhật  $h_{index}$  của mình và của hàng xóm khi nhận được gói tin Acknowledgement trong quá trình truyền tin (cách cập nhật  $h_{index}$  sẽ được trình bày ở bên dưới).

### (iii) Năng lượng còn lại của các nút mạng

Nút mạng có năng lượng cao sẽ được ưu tiên chọn làm next hop để gửi gói tin tới.

### (iv) Tần suất gói tin đã gửi đi

Để cân bằng năng lượng cho các hàng xóm, mỗi nút mạng sẽ ưu tiên chọn hàng xóm có tần suất gửi tin từ nó tới hàng xóm đó nhỏ nhất trong số tất cả các hàng xóm để làm next hop.

Theo nguyên lý của phương pháp học tăng cường, mỗi nút mạng sau khi nhận được một gói tin dữ liệu, nó cần gửi lại một gói tin Acknowledgement trong đó chứa các thông tin về reward, để nút gửi cập nhật bảng Q-value. Tuy nhiên vì mạng có rất nhiều cảm biến, trong đó có rất nhiều nút nguồn, mỗi nguồn lại gửi một lượng lớn gói tin dữ liệu tới đích, do đó số lần gói tin dữ liệu được gửi đi là rất lớn. Nếu gói tin Acknowledgement được gửi lại sau mỗi lần nhận gói tin dữ liệu thì không những không đảm bảo được cân bằng tải cho mạng mà còn làm cho các nút mạng bị cạn kiệt năng lượng do gửi quá nhiều gói tin Acknowledgement, thậm chí gây ra xung đột trong mạng khiến các gói tin dữ liệu không thể gửi được tới đích.

Vì lý do trên, để tránh tiêu hao năng lượng của các nút mạng và tránh xung đột gói tin, giao thức định tuyến của chúng tôi chỉ thực hiện gửi lại gói tin Acknowledgement theo định kỳ. Cụ thể khi năng lượng còn lại của nút nhận thỏa mãn điều kiện sau thì nó sẽ gửi lại gói tin Acknowledgement cho nút nhận:

$$\text{Remaining Energy} < \text{Initial Energy} * \text{Energy Reduction Threshold}$$

Trong 4 yếu tố (i) (ii) (iii) (iv) đã nêu trên, có 2 yếu tố *Tần suất gói tin đã gửi* và *Khoảng cách Ó-clit tới đích* có thể được cập nhật ngay tại nút gửi sau khi đã truyền gói tin đi, 2 yếu tố còn lại *Khoảng cách tới hồ của các nút mạng trên đường định tuyến* và *Năng lượng còn lại của các nút mạng* thì cần cập nhật từ gói tin Acknowledgement. Chúng tôi sẽ chia phần cập nhật Q-value thành 2 phần:

1. Cập nhật Q-value của các hàng xóm ngay sau khi gửi gói tin dữ liệu đi. Hàm reward được thiết kế cho phần cập nhật này phụ thuộc vào khoảng cách Ó-clit tới hồ của nút nhận và tần suất gửi gói tin tới nút nhận (tần suất này được tính bằng hàm số của số lần nút gửi đã gửi tin dữ liệu tới nút nhận). Công thức tính

của hàm reward như sau:

$$reward = \begin{cases} -10, & \text{nếu không tìm được next hop} \\ 15, & \text{nếu next hop là đích} \\ \beta_1 * adv/adv_{avg} + (1 - \beta_1) * \frac{1}{1 + e^{h_{sendFrequency}}}, & \\ \text{trong các trường hợp còn lại} \end{cases} \quad (4.1)$$

trong đó,

$adv = distance(i, k) - distance(n, k)$  là hiệu khoảng cách từ nút gửi i tới đích k và khoảng cách từ nút nhận n tới đích k,

$adv_{avg} = \sum(adv)/m$  với m là tổng số hàng xóm của nút hiện tại

$$sendFrequency = \varphi * \frac{1}{NOW - t} + (1 - \varphi) * oldSendFrequency \quad (4.2)$$

Trong công thức 4.2 ở trên, NOW là thời điểm gửi gói tin hiện tại, t là thời điểm gửi gói tin lần gần nhất trước đó;  $\varphi$  là hằng số quyết định độ quan trọng của tần suất gửi tin tức thời và tần suất gửi tin trong quá khứ,  $\varphi$  lớn thể hiện tần suất tức thời là quan trọng,  $\varphi$  nhỏ thể hiện tần suất trong quá khứ là quan trọng.

2. Cập nhật Q-value của các hàng xóm khi nhận được gói tin Acknowledgement. Công thức tính của hàm reward như sau:

$$reward = \begin{cases} -10, & \text{nếu không tìm được next hop} \\ 15, & \text{nếu next hop là đích} \\ -\beta_2 * (1 - \frac{E_{residual}}{E_{initial}}) - (1 - \beta_2) * \frac{1}{1 + e^{h_{index}}}, & \\ \text{trong các trường hợp còn lại} \end{cases} \quad (4.3)$$

trong đó,  $E_{residual}$  là năng lượng còn lại của nút nhận,  $E_{initial}$  là năng lượng khởi tạo của nút nhận,  $h_{index}$  là chỉ số thể hiện khoảng cách tới hố của nút nhận (đã được định nghĩa ở trên). Hàm reward tính theo công thức 4.3 này cần thông tin từ nút nhận, các thông tin này được gắn vào header của gói tin Acknowledgement. Có 2 loại gói tin Acknowledgement là *gói tin Acknowledgement đặc biệt*: được gửi lại từ một stuck node hoặc từ nút đích; và *gói tin Acknowledgement thông thường* được gửi lại từ các nút không phải stuck node cũng không phải nút đích. Các thông tin được gắn vào 2 loại gói tin này như sau:

Field	Type
<i>Residual Energy</i>	double
<i>V Value</i>	double
<i>h<sub>index</sub></i>	int

(a) Gói tin thông thường

Field	Type
<i>Reward (constant reward)</i>	double
<i>V Value</i>	double

(b) Gói tin đặc biệt

Bảng 4.1: Header của gói tin Acknowledgement

Sau khi tính được reward, Q-value của hàng xóm sẽ được cập nhật theo công thức:

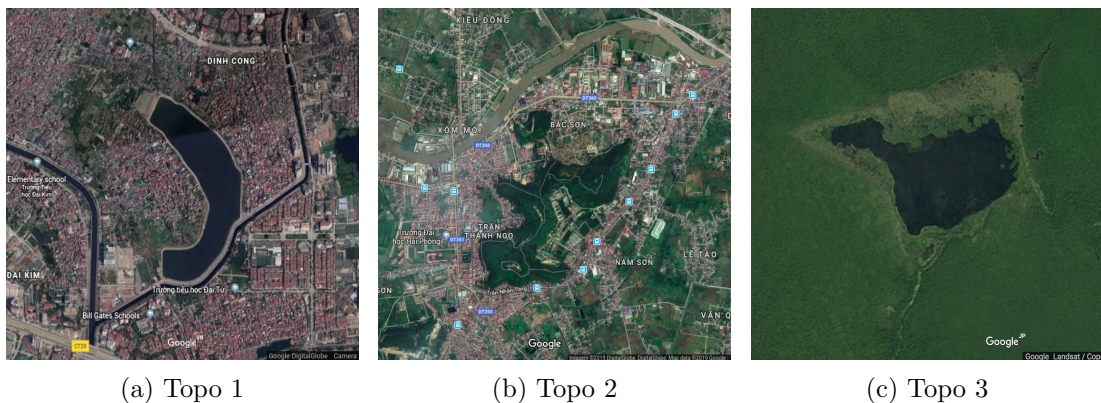
$$Q(s, a) \leftarrow (1 - \alpha).Q(s, a) + \alpha.[r + \gamma.\max_{a'} Q(s', a')] \quad (4.4)$$

Trong đó,  $\alpha$  là learning rate, r là reward và  $\gamma$  là discount factor,  $\max_{a'} Q(s', a')$  chính là V value của nút nhận (chương 3.3).

## Chương 5

# Thí nghiệm và đánh giá hiệu năng của giao thức đề xuất

### 5.1 Kịch bản thí nghiệm



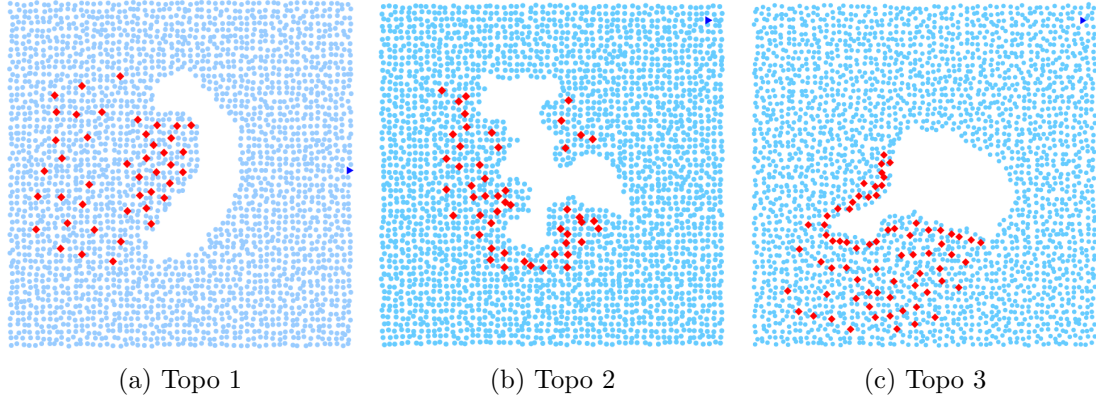
Hình 5.1: Danh sách địa hình mạng lấy từ bản đồ của Google Maps.

Chúng tôi chọn 3 địa hình thực tế (hình 6.1) lấy từ dữ liệu của Google Maps để làm thực nghiệm. Những địa hình này là vùng có hồ nước, nơi không thể đặt được cảm biến. Khi định tuyến, các hồ nước này trở thành hố định tuyến làm cản trở đường đi của gói tin. Trong các địa hình (a)(b)(c) ở trên, phần màu xanh đậm (topo 2) và màu đen đậm (topo 1, 3) ở chính giữa của các địa hình là các hồ nước.

Để chuyển các địa hình trong hình 6.1 thành mô hình mạng, chúng tôi sử dụng công cụ tạo kịch bản mạng và phân tích kết quả thí nghiệm **WiSSim** [28]. Công cụ WiSSim cho phép đưa tệp dưới dạng ảnh vào để làm khung nền cho mạng. Sau khi file ảnh của địa hình thực tế được đưa vào để làm khung nền, chúng tôi tạo mạng có kích thước  $1000 \times 1000 m^2$ , trong đó 3000 nút mạng được phân bố trong mạng bằng cách chia mạng thành lưới  $55 \times 55$  ô vuông, và phân bố 1 nút mạng ở vị trí ngẫu nhiên trong mỗi ô vuông ấy. Mật độ của mạng là  $1 \text{ node} / 18 \times 18 m^2$ . Sau đó nút mạng ứng với vị trí của các hồ nước trong khung nền sẽ được xóa đi để tạo thành hố.

Cả ba topo mạng đều được bố trí duy nhất một nút đích (base station) ở biên mạng

hoặc góc mạng. Topo 1, topo 2 và topo 3 lần lượt có 44, 51 và 72 nút nguồn tương ứng nằm ở vùng lặn cận hố (Hình 6.3). Thời gian mô phỏng thí nghiệm là 3000s. Thông số của các nút mạng được thể hiện ở Bảng 6.1.



Hình 5.2: Vị trí nguồn đích của các topo mạng. Các nút màu xanh nhạt là các nút mạng, các vị trí màu đỏ hình thoi là nút nguồn, vị trí màu xanh đậm hình tam giác là đích.

Factor	Value
MAC type	CSMA/CA
Interface queue model	DropTail
Transmission of radio	TwoRayGround
Antenna type	OmniAntenna
Node initial energy	30 J
Queue length	50 packets
Transmission range	40 m
Node idle power	9.6 mW
Node receive power	45 mW
Node transmit power	88.5 mW
Packet sending interval	10 s
Data packet size	50 bytes

(a) Thông số của một nút mạng

Perimeter	Value
Learning Rate $\alpha$	0.9
Discount Factor $\gamma$	0.5
$\beta_1$	1
$\beta_2$	2
$\varphi$ (send frequency factor in equation 4.2)	0.5
Energy Reduction Threshold	0.5

(b) Thông số của giao thức HYBRID. Các tham số này được xác định dựa trên nhiều lần thí nghiệm và chọn ra bộ số cho kết quả tốt nhất

Bảng 5.1: Thông số thí nghiệm

## 5.2 Các tiêu chí đánh giá hiệu năng

Chúng tôi so sánh hiệu năng của giao thức do chúng tôi đề xuất - HYBRID, với hai giao thức đã được đề xuất trước đây: BSMH [19] và EDGR [20], trong đó giao thức BSMH được thí nghiệm với 2 giá trị của  $\epsilon$  là 0.3 và 0.8, giao thức của chúng tôi có các thông số được trình bày ở Bảng 6.2. Các chỉ số được dùng để đánh giá hiệu năng bao gồm: **Thời gian sống của mạng** (*Network lifetime*), **Thời gian gói tin đi từ nguồn tới đích** (*Delay*), **Tỉ lệ gói tin tới đích thành công** (*Delivery ratio*), **Năng lượng trung bình để gửi một gói tin** (*Energy consumption per packet*). Ý nghĩa và cách tính của mỗi chỉ số như sau:



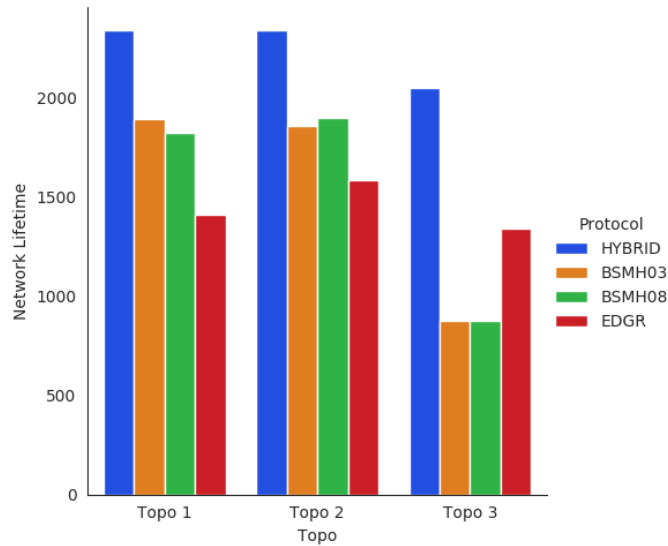
1. *Thời gian sống của mạng*: Chỉ số này được dùng trong hầu hết tất cả các bài báo đề xuất giao thức truyền tin. Có thể thấy, khi có một nút trong mạng chết đi, đặc biệt là nút đó nằm ở vị trí trung tâm của các đường định tuyến, sẽ có rất nhiều gói tin khi được truyền đến nút đó thì bị hủy và không tới được đích, làm giảm hiệu quả truyền tin của mạng. Thời gian sống của mạng được tính từ thời điểm mạng bắt đầu hoạt động cho tới thời điểm nút đầu tiên trong mạng chết đi.
2. *Thời gian gói tin đi từ nguồn tới đích*: Chúng tôi tính trung bình cộng thời gian đi từ nguồn tới đích của tất cả các gói tin tới đích thành công. Thời gian gói tin đi từ nguồn tới đích phụ thuộc vào 3 yếu tố: (i) *số nút mà gói tin đã đi qua* (hay nói cách khác là *độ dài đường đi của gói tin*) - số nút này càng ít, gói tin đi tới đích càng nhanh; (ii) *độ đơn giản của giao thức* - giao thức càng đơn giản, thời gian tính càng nhanh, do đó gói tin sẽ chọn được next hop càng nhanh; (iii) *cân bằng tải của giao thức* - nếu giao thức có độ cân bằng tải không tốt, sẽ có một bộ phận các nút phải nhận nhiều gói tin hơn các nút khác, dẫn đến tình trạng các gói tin được gửi tới những nút đó sẽ phải xếp hàng đợi để chờ các gói tin trước đó được gửi đi, và làm kéo dài thời gian đi tới đích.
3. *Tỷ lệ gói tin tới đích thành công*: Được tính bằng tỷ lệ số gói tin tới đích thành công chia cho số gói tin đã được gửi đi từ nút nguồn.
4. *Năng lượng trung để gửi một gói tin*: Được tính bằng tổng năng lượng đã tiêu hao của tất cả các nút chia cho tổng số gói tin tới đích thành công. Một giao thức định tuyến hiệu quả sẽ có năng lượng tiêu hao của các nút mạng thấp và tổng số gói tin tới đích nhiều.

## 5.3 Phân tích kết quả thí nghiệm

### 5.3.1 Thời gian sống của mạng

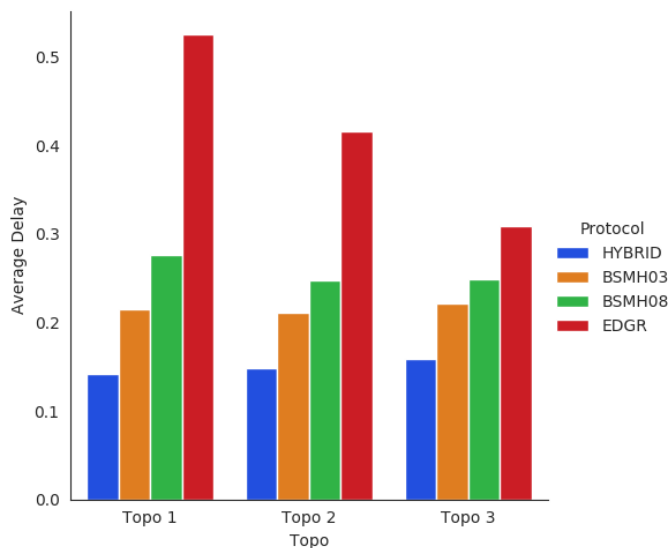
Kết quả thí nghiệm thời gian sống của mạng (network lifetime) được thể hiện ở Hình 5.1. Có thể thấy network lifetime của giao thức HYBRID do chúng tôi đề xuất đạt kết quả cao (lớn hơn 1.2 lần so với giao thức BSMH ở Topo 1, 2 và lớn hơn 1.5 lần so với giao thức EDGR ở Topo 3). Đồng thời, network lifetime của giao thức HYBRID cũng ổn định ở cả 3 Topo mạng. Trong khi đó giao thức BSMH đạt kết quả cao hơn EDGR ở Topo 1 và Topo 2 nhưng lại thấp hơn EDGR ở Topo 3. Giao thức BSMH xác định sẵn các đích tạm thời và cho các gói tin đi theo greedy đến cách đích tạm thời ấy. Đường định tuyến của giao thức này chỉ đảm bảo khác nhau qua các lần đi (nhờ thay đổi tỷ lệ vị tự) mà không quan tâm đến năng lượng còn lại của các nút mạng. Giao thức EDGR có quan tâm đến năng lượng còn lại của các nút mạng nhưng lại chỉ tối ưu địa phương, tức là chỉ đảm bảo cân bằng tải giữa các hàng xóm của một nút mạng, và vẫn cho gói tin đi theo greedy tới các đích tạm thời. Khác với hai giao thức này, giao

thức của chúng tôi sử dụng Q-learning để thực hiện back propagation về trạng thái năng lượng của cả những nút mạng ở xa, đồng thời nút mạng sẽ liên tục học và cập nhật được trạng thái về năng lượng của lẫn nhau, vì vậy cân bằng tải của HYBRID tốt hơn, dẫn đến network lifetime được kéo dài. Ngoài ra, việc tính toán phức tạp và phát tán nhiều thông tin về hồ (của giao thức BSMH) hay gửi gói tin BURST (của giao thức EDGR) cũng làm tiêu hao năng lượng của các nút mạng.



Hình 5.3: Kết quả thí nghiệm thời gian sống của mạng

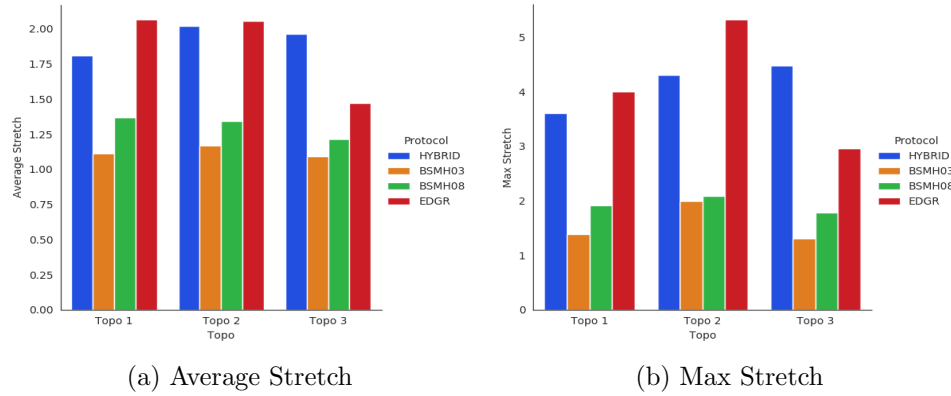
### 5.3.2 Thời gian gói tin đi từ nguồn tới đích



Hình 5.4: Kết quả thí nghiệm trung bình thời gian gói tin đi từ nguồn tới đích

Qua kết quả trên, giao thức HYBRID cho thấy hiệu quả trong tốc độ truyền tin. Thời gian truyền tin từ nguồn tới đích của HYBRID chỉ bằng khoảng 0.65 lần thời gian truyền tin của BSMH và khoảng 0.26 lần thời gian truyền tin của EDGR. Như đã đề

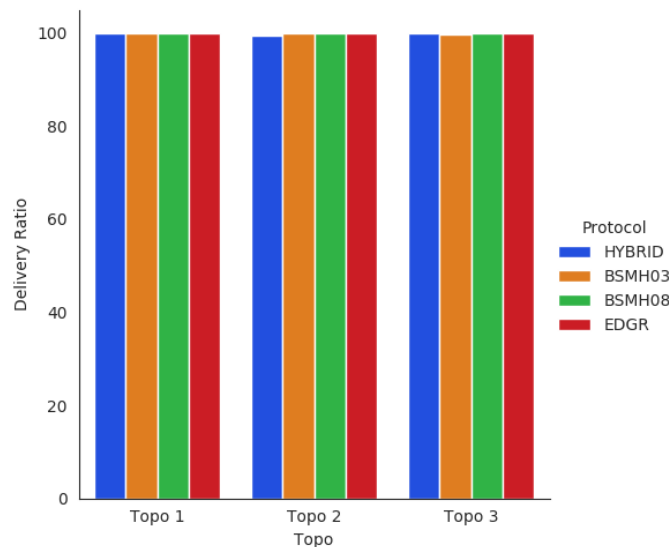
cập ở trên, tốc độ truyền tin phụ thuộc vào 3 yếu tố là: độ dài đường đi, độ phức tạp tính toán, và cân bằng tải. Khi đánh giá tốc độ truyền tin, chúng tôi cũng đã thí nghiệm đo kết quả về độ dài đường đi của ba giao thức, được thể hiện ở hai chỉ số *Average Stretch* và *Max Stretch* được thể hiện ở hai đồ thị dưới đây:



Hình 5.5: Kết quả thí nghiệm về độ dài đường định tuyến

Nhìn vào đồ thị trên ta có thể thấy rằng độ dài đường đi của giao thức HYBRID không tốt hơn độ dài đường đi của giao thức EDGR. Cụ thể là ở Topo 1 và Topo 2 thì HYBRID tốt hơn EDGR nhưng tỷ lệ đường đi giữa 2 giao thức chỉ rơi vào khoảng 0.8, và Topo 3 thì HYBRID lại có độ dài đường đi gấp khoảng 1.2 lần độ dài đường đi của EDGR. Tuy nhiên, tốc độ truyền tin của HYBRID vẫn tốt hơn EDGR rất nhiều bởi HYBRID đã làm tốt được 2 yếu tố là: độ đơn giản tính toán và cân bằng tải. Xét về BSMH, BSMH luôn cho kết quả về độ dài đường đi rất tốt ở cả 3 topo, tuy nhiên giao thức này có độ phức tạp tính toán khá lớn, dẫn đến tốc độ truyền tin không tốt.

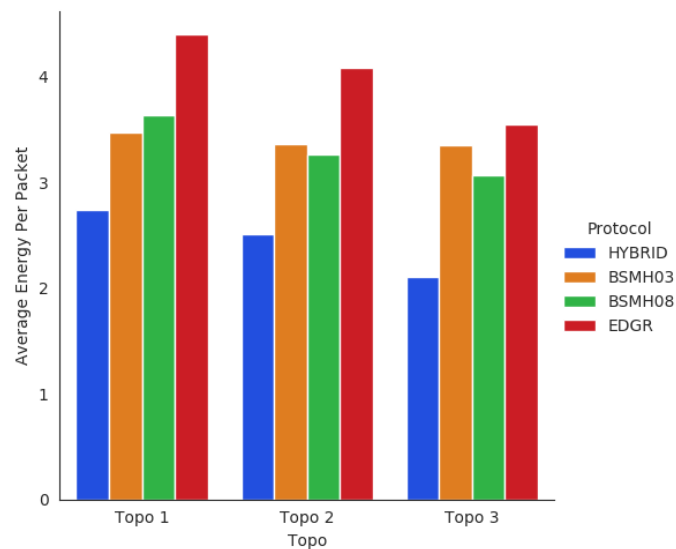
### 5.3.3 Tỷ lệ gói tin tới đích thành công



Hình 5.6: Kết quả thí nghiệm tỷ lệ gói tin tới đích thành công

Tỷ lệ gói tin tới đích thành công của cả ba giao thức đều đạt xấp xỉ 100% ở cả ba topo mạng. Ở Topo 2, tỷ lệ gửi tin thành công của HYBRID thấp hơn một chút so với 2 giao thức còn lại (99.33%). Bởi các đích tạm thời được xác định trước khi truyền tin và gắn vào header của gói tin dữ liệu, hai giao thức BSMH và EDGR đều được chứng minh dựa trên lý thuyết là luôn tìm được next hop. Vì muốn tiết kiệm năng lượng và thời gian ở khâu tính toán đường đi trước khi truyền tin, đồng thời giảm lượng thông tin mà gói tin dữ liệu phải mang đi, giao thức HYBRID của chúng tôi chỉ định tuyến dựa trên kinh nghiệm trong quá khứ và thông tin của các nút hàng xóm. Tuy không có chứng minh lý thuyết về việc đảm bảo tỷ lệ truyền tin, kết quả thực nghiệm cho thấy HYBRID có tỷ lệ truyền tin thành công trên 99% ở cả 3 topo mạng.

### 5.3.4 Năng lượng trung bình để gửi một gói tin

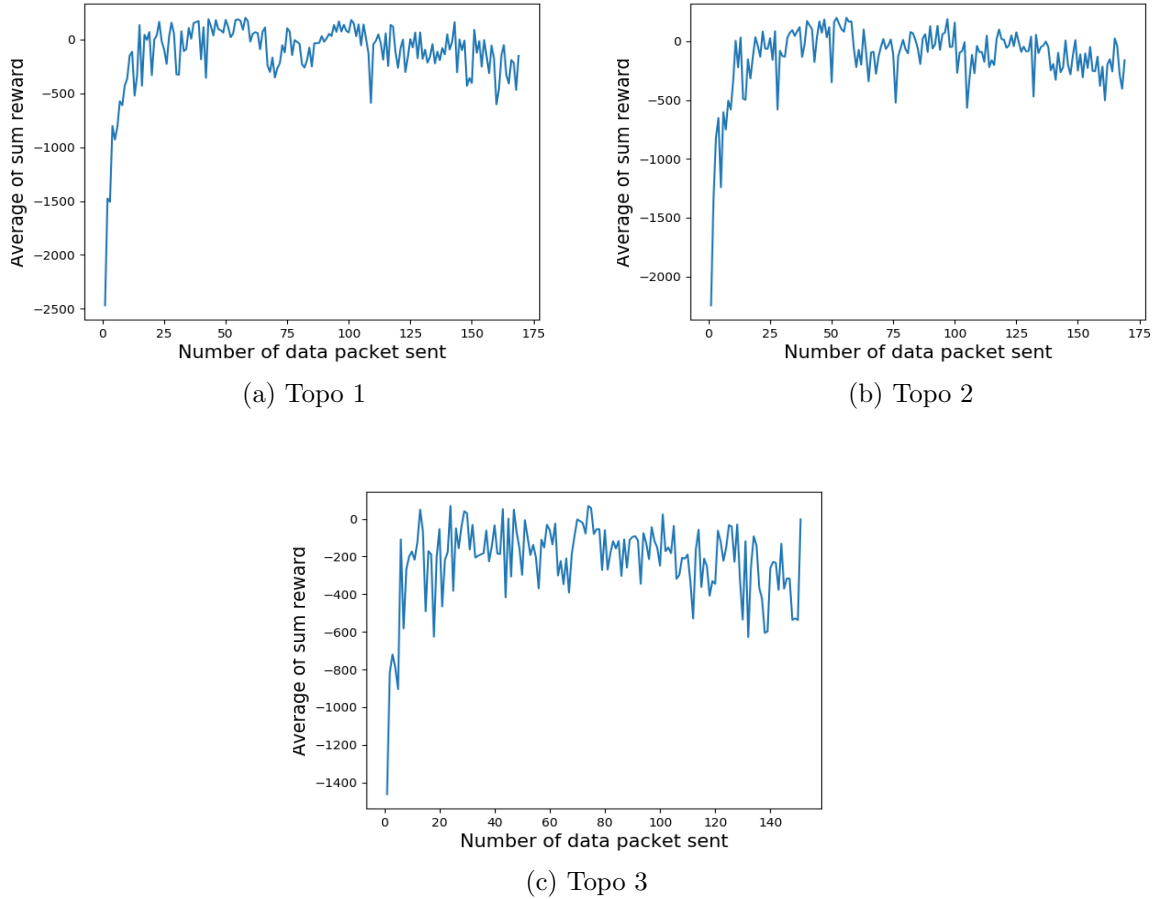


Hình 5.7: Kết quả thí nghiệm năng lượng tiêu hao trung bình cho mỗi gói tin gửi tới đích

Giao thức HYBRID cho kết quả tốt ở cả ba topo mạng, BSMH cho kết quả tốt thứ 2, và giao thức EDGR luôn cho kết quả tồi nhất. Nhìn vào kết quả số gói tin tới đích thành công ở chương 5.3.3, chúng ta kết luận được rằng số lượng gói tin tới đích thành công của cả ba giao thức là tương đương nhau. Vì vậy năng lượng tiêu hao trung bình cho mỗi gói tin gửi tới đích cao hay thấp sẽ phụ thuộc vào 2 yếu tố: *Tổng năng lượng tiêu hao của mỗi nút mạng* và *Số lượng nút mạng đã tham gia vào quá trình định tuyến*. Nhìn vào kết quả thí nghiệm về độ dài đường đi ở Hình 5.7, ta có thể nói rằng EDGR luôn có năng lượng tiêu hao trung bình của mỗi gói tin tồi nhất vì đường đi của giao thức này dài nhất. Độ dài đường đi của HYBRID lớn hơn độ dài đường đi của BSMH chứng tỏ có nhiều nút mạng tham gia vào quá trình định tuyến hơn, nhưng nhờ quá trình tính toán đường đi đơn giản và dung lượng thông tin chứa trong header của gói tin ít hơn (cụ thể, dựa vào các thông tin chứa trong header, chúng tôi đã tính ra dung lượng thông tin mà gói tin dữ liệu của mỗi giao thức HYBRID, BSMH, EDGR cần truyền đi lần lượt là: 24 bytes, 300 bytes, 308 bytes), HYBRID đã tiết kiệm được

năng lượng truyền và nhận tin cho mỗi nút mạng.

### 5.3.5 Tổng reward trung bình theo thời gian của các gói tin trong giao thức HYBRID



Hình 5.8: Trung bình của tổng Reward của các gói tin dữ liệu theo thời gian của từng topo.

Theo định nghĩa về học tăng cường ở chương 3.1, mục tiêu của quá trình học có trạng thái kết thúc là tối ưu lượng  $R = r_1 + r_2 + \dots + r_n$ . Khi áp dụng học tăng cường vào định tuyến, chúng tôi coi trạng thái kết thúc của gói tin dữ liệu là khi nó được truyền tới đích. Chúng tôi đánh giá lượng  $R = r_1 + r_2 + \dots + r_n$  bằng cách đo tổng reward được gửi về khi gói tin đi hết một lượt từ nguồn tới đích của các gói tin xuất phát từ các nguồn khác nhau, rồi lấy trung bình của các kết quả ấy. Ở cả 3 topo, với thời gian mô phỏng là 3000s, có tất cả 175 gói tin dữ liệu được gửi đi từ mỗi nút nguồn. Trung bình của tổng reward của các gói tin dữ liệu theo thời gian được thể hiện ở Hình 5.7. Qua đồ thị này, chúng ta có thể thấy thời gian để giá trị reward trở nên ổn định, còn gọi là thời gian cần thiết để xây dựng được một bảng Q-value ổn định hay *convergence time* rơi vào khoảng 20 gói tin đầu tiên. Điều này cho thấy rằng, chỉ sau khoảng 10% thời gian đầu tiên, các gói tin đã có thể đi theo đường đi "tối ưu" từ nguồn tới đích.

# Chương 6

## Kết luận

### 6.1 Kết quả đạt được

Dựa trên các ưu điểm và nhược điểm của các giao thức định tuyến địa lý tránh hố trước đây, trong đề án này, chúng tôi đã đề xuất giao thức định tuyến tránh hố HYBRID kết hợp giữa định tuyến địa lý và phương pháp Q-learning. Kết quả thí nghiệm cho thấy:

- Giao thức của chúng tôi đã kéo dài thời gian sống của mạng thêm 20% so với các giao thức trước đây.
- Tốc độ truyền tin của giao thức do chúng tôi đề xuất nhanh gấp 1.5 lần so với các giao thức trước đây.
- Tỷ lệ truyền tin tới đích thành công đạt hơn 99%.

Với kết quả trên, chúng tôi nhận định rằng áp dụng học tăng cường vào định tuyến trong mạng cảm biến không dây là một hướng tiếp cận phù hợp. Và trong quá trình nghiên cứu, tiến hành cài đặt thí nghiệm, chúng tôi nhận ra còn có nhiều hướng để mở rộng và cải tiến việc áp dụng học tăng cường, cụ thể chúng tôi sẽ trình bày ở chương [6.2](#) dưới đây.

### 6.2 Hướng phát triển

Chúng tôi có hai hướng phát triển giao thức như sau:

1. Tuy giao thức HYBRID đã được thiết kế để gói tin có thể định tuyến được trong các địa hình mạng có một hay nhiều hố, nhưng trong khuôn khổ của đề án này, chúng tôi mới chỉ thí nghiệm đối với những topo mạng có một hố duy nhất. Để giao thức được toàn diện, chúng tôi hướng tới việc thí nghiệm đánh giá và tối ưu giao thức cho các trường hợp mạng *có nhiều hố, hoặc địa hình hố thay đổi theo thời gian*.
2. Phương pháp Q-learning thuộc nhóm thuật toán *Tabular Solution Methods*, tức là chỉ phù hợp với những đối tượng có không gian của tập *action* (hành động)

và tập *state* (trạng thái) là nhỏ. Vì hiện tại chúng tôi hướng tới các kịch bản mạng có số ít nút đích (ít base station) nên chúng tôi đã chọn lựa phương pháp Q-learning. Trong tương lai, để giao thức có thể hoạt động tốt trong trường hợp mạng có rất nhiều nút đích, chúng tôi sẽ thay đổi cách sử dụng phương pháp học tăng cường, ví dụ như chuyển sang dùng nhóm thuật toán *Approximation Solution Methods* thay vì *Tabular Solution Methods*.

# Tài liệu tham khảo

- [1] [https://vi.wikipedia.org/wiki/C%E1%BA%A3m\\_bi%E1%BA%BFn](https://vi.wikipedia.org/wiki/C%E1%BA%A3m_bi%E1%BA%BFn)
- [2] S. H. Lee, S. Lee, H. Song, and H. S. Lee, “Wireless sensor network design for tactical military applications : Remote large-scale environments,” in *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pp. 1–7, Oct 2009.
- [3] E. Onur, C. Ersoy, H. Delic, and L. Akarun, “Surveillance wireless sensor networks: Deployment quality analysis,” *IEEE Network*, vol. 21, pp. 48–53, November 2007.
- [4] D. Chen, Z. Liu, L. Wang, M. Dou, J. Chen, and H. Li, “Natural disaster monitoring with wireless sensor networks: A case study of data-intensive applications upon low-cost scalable systems,” *Mob. Netw. Appl.*, vol. 18, pp. 651–663, Oct. 2013.
- [5] M. Erdelj, M. Król, and E. Natalizio, “Wireless sensor networks and multi-uav systems for natural disaster management,” *Computer Networks*, vol. 124, pp. 72 – 86, 2017.
- [6] G. Han, X. Yang, L. Liu, M. Guizani, and W. Zhang, “A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks,” *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2017.
- [7] A. S. Bhosle and L. M. Gavhane, “Forest disaster management with wireless sensor network,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 287–289, March 2016.
- [8] E. Cayirci and T. Coplu, “Sendrom: Sensor networks for disaster relief operations management,” *Wireless Networks*, vol. 13, pp. 409–423, Jun 2007.
- [9] A. ur Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, “A review of wireless sensors and networks’ applications in agriculture,” *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 263 – 270, 2014.
- [10] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, “Sensorscope: Application-specific sensor network for environmental monitoring,” *ACM Transaction on Sensor Network*, vol. 6, pp. 17:1–17:32, Mar. 2010..



- [11] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66 – 84, 2015.
- [12] X. Shi, L. Fan, J. Xia, Z. Tang, and H. Li, "An environment monitoring system for precise agriculture based on wireless sensor networks," in *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks(MSN)*, vol. 00, pp. 28–35, 12 2011.
- [13] Y. Zhu, J. Song, and F. Dong, "Applications of wireless sensor network in the agriculture environment monitoring," *Procedia Engineering*, vol. 16, pp. 608 – 614, 2011.
- [14] M. M. Baig and H. Gholamhosseini, "Smart health monitoring systems: An overview of design and modeling," *Journal of Medical Systems*, vol. 37, pp. 98 – 98, Jan 2013.
- [15] M. R. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators A: Physical*, vol. 162, no. 1, pp. 116 – 129, 2010.
- [16] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688 – 2710, 2010.
- [17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.
- [18] Phi-Le Nguyen, Yusheng Ji, Khanh Le, Thanh-Hung Nguyen, "Load balanced and constant stretch routing in the vicinity of holes in WSNs", *15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018.
- [19] Phi-Le Nguyen, Yusheng Ji, Khanh Le, Thanh-Hung Nguyen, "Routing in the Vicinity of Multiple Holes in WSNs," *2018 5th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, 2018.
- [20] Haojun Huang, Hao Yin, Geyong Min, Junbao Zhang, Yulei Wu, Xu Zhang, "Energy-Aware Dual-Path Geographic Routing to Bypass Routing Holes in Wireless Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 17, issue 6, 2018.
- [21] Fucui Yu, Shengli Pan, and Guangmin Hu, "Hole Plastic Scheme for Geographic Routing in Wireless Sensor Networks", *IEEE ICC - Ad-hoc and Sensor Networking Symposium*, 2015.
- [22] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, pp. 28–34, Oct 2000.

- [23] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM'04, vol. 4, pp. 2458–2468 vol.4, March 2004.
- [24] Wen-Jiunn Liu and Kai-Ten Feng, "Greedy Routing with Anti-Void Traversal for Wireless Sensor Networks", *IEEE TRANSACTIONS ON MOBILE COMPUTING*, VOL.8, NO.7, JULY 2009.
- [25] Zhiping Kang, Honglin Yu, and Qingyu Xiong, "Detection and Recovery of Coverage Holes in Wireless Sensor Networks", 2013.
- [26] Phi Le Nguyen, Van Khanh Nguyen, "On hole approximation algorithms in wireless sensor networks", *Journal of Computer Science and Cybernetics*, Vol.30, No.4, 2014.
- [27] Brad Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proceeding MobiCom 00 Proceedings of the 6th annual international conference on Mobile computing and networking*, Pages 243-254, 2000.
- [28] <http://sedic.soict.hust.edu.vn/wissim/>.
- [29] <https://www.isi.edu/nsnam/ns/>
- [30] D.T. Lee, F.P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers", *Networks*, vol. 14, no. 3, pp. 393-410, 1984.
- [31] S. Subramanian, S. Shakkottai, and P. Gupta, "On optimal geographic routing in wireless networks with holes and non-uniform traffic," in *Proc. of the 26th IEEE International Conference on Computer Communications*, INFOCOM'07, pp. 1019–1027, May 2007.
- [32] M. Choi, H. Choo, "Bypassing Hole Scheme Using Observer Packets for Geographic Routing in WSNs", *The International Conference on Information Networking*, 2011 (ICOIN2011).
- [33] F. Yu et al, "Efficient Hole Detour Scheme for Geographic Routing in Wireless Sensor Networks", In *Proc. of the 67th IEEE Vehicular Technology Conference*, VTC'08, pages 153-157, 2008.
- [34] Y. Tian et al., "Energy-Efficient Data Dissemination Protocol for Detouring Routing Holes in Wireless Sensor Networks". In *Proc. of IEEE Intl. Conf. on Communications*, ICC'08, pages 2322-2326, 2008.
- [35] Phi-Le Nguyen, Duc-Trong Nguyen, Khanh-Van Nguyen, "Load balanced routing with constant stretch for wireless sensor network with holes", *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014.

- [36] Shaoqiang Dong, Prathima Agrawal and Krishna Sivalingam, "Reinforcement Learning Based Geographic Routing Protocol for UWB Wireless Sensor Network", *Communications Society subject matter experts for publication* in the IEEE GLOBECOM 2007.
- [37] Ping Wang, Ting Wang, "Adaptive Routing for Sensor Networks using Reinforcement Learning", Proceedings of The Sixth IEEE *International Conference on Computer and Information Technology* (CIT'06), 2006.
- [38] Tiansi Hu, Yungsi Fei, "QELAR: A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks", Published by the IEEE CS, CASS, *ComSoc*, 2010.