ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG

--- --- ---

BÁO CÁO LẬP TRÌNH MẠNG

<u>Tên đề tài</u> : HỆ THỐNG ĐIỀU KHIỂN DRONE GIAO HÀNG TỰ ĐỘNG

Sinh viên thực hiện:	Mã số sinh viên
Lê Duy Tân	106220269
Nguyễn Văn Mùi	106220261
Ngô Thành Trung	106220239

Giảng viên hướng dẫn: Nguyễn Văn Hiếu

Đà Nẵng , tháng 11 năm 2025

MỤC LỤC

LỜI NÓI ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang phát triển mạnh mẽ, công nghệ drone (thiết bị bay không người lái) đã và đang trở thành một trong những giải pháp tiên tiến cho nhiều lĩnh vực như giám sát, khảo sát địa hình, nông nghiệp thông minh và đặc biệt là logistics - giao hàng tự động. Các tập đoàn công nghệ lớn như Amazon, Google, DHL đều đang đầu tư nghiên cứu và triển khai hệ thống giao hàng bằng drone nhằm tối ưu hóa chi phí, rút ngắn thời gian vận chuyển và giảm thiểu tác động môi trường.

Để xây dựng một hệ thống drone giao hàng hoạt động hiệu quả, bên cạnh các kỹ thuật điều khiển bay, định vị GPS và xử lý cảm biến, lập trình mạng đóng vai trò then chốt trong việc thiết lập kênh truyền thông tin giữa drone và trạm điều khiển mặt đất (Ground Control Station - GCS). Việc truyền nhận dữ liệu telemetry (vị trí, tốc độ, độ cao, trạng thái pin), gửi lệnh điều khiển bay và giám sát thời gian thực đều dựa trên các giao thức mạng như TCP/IP, UDP và MAVLink (Micro Air Vehicle Link).

Đề tài không chỉ giúp củng cố kiến thức về lập trình mạng (bao gồm các khái niệm socket, giao thức TCP/UDP, mô hình client-server, blocking/non-blocking I/O) mà còn mở rộng khả năng ứng dụng vào lĩnh vực công nghệ cao - điều khiển hệ thống bay tự động. Đây là bước đệm quan trọng để tiếp cận các bài toán thực tế phức tạp hơn như điều khiển đội drone (swarm control), tối ưu hóa đường bay và tích hợp trí tuệ nhân tạo vào hệ thống giao hàng.

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Văn Hiếu, người đã trực tiếp hướng dẫn và hỗ trợ chúng em trong quá trình thực hiện báo cáo này. Sự quan tâm, tận tâm và sự đồng hành của thầy đã giúp chúng em vượt qua những khó khăn, phát triển kiến thức và kỹ năng của mình một cách toàn diện.

Chúng em cũng nhận thức rằng bài báo cáo này vẫn còn hạn chế và có thể chưa hoàn thiện hoàn toàn. Vì vậy, chúng em rất mong nhận được sự đánh giá và góp ý thêm từ thầy để chúng em có thể tiếp tục hoàn thiện và phát triển kỹ năng của mình. Chúng em tin tưởng vào sự am hiểu và sự tận tâm của thầy, và hy vọng sẽ tiếp tục được học hỏi và phát triển dưới sự hướng dẫn của thầy trong tương lai.

Một lần nữa, chúng em xin chân thành cảm ơn thầy Nguyễn Văn Hiếu vì những đóng góp và sự hỗ trợ của thầy trong quá trình thực hiện báo cáo này. Sự giúp đỡ của thầy đã góp phần quan trọng vào sự thành công của chúng em. Chúng em rất biết ơn và trân trọng sự tận tâm và kiến thức mà thầy đã truyền đạt cho chúng em.

Chúng em xin chân thành cảm ơn!

BẢNG PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

STT	HỌ VÀ TÊN	NHIỆM VỤ	KHỐI LƯỢNG
01	NGÔ THÀNH TRUNG	Lý thuyết (Chương 2) Thiết kế (Chương 3) Triển khai (Chương 3/4): Slide	33%
02	LÊ DUY TÂN	Lý thuyết (Chương 2) Thiết kế (Chương 3) Triển khai (Chương 3/4) Slide	33%
03	NGUYỄN VĂN MÙI	Tổng quan (Chương 1) Kịch bản (Chương 3) Đánh giá (Chương 4/5) Slide	33%

CHƯƠNG 1. TỔNG QUAN

1.1. Tên đề tài:

Xây dựng hệ thống điều khiển drone giao hàng tự động

1.2. Giới thiệu:

Đề tài: "Xây dựng hệ thống điều khiển drone giao hàng tự động" tập trung vào việc thiết kế và triển khai một hệ thống mạng cho phép giám sát, điều khiển và giao tiếp với drone thông qua giao thức mạng. Hệ thống được xây dựng trên nền tảng PX4 Autopilot, mô phỏng Gazebo và ứng dụng điều khiển QGroundControl (QGC), đồng thời tích hợp các kỹ thuật lập trình mạng để thiết lập kênh truyền thông tin giữa trạm điều khiển mặt đất (Ground Control Station - GCS) và drone.

Hệ thống drone giao hàng là một ứng dụng điển hình của mô hình **Client-Server** và yêu cầu truyền thông tin thời gian thực với độ tin cậy cao. Các thành phần chính bao gồm:

- **Drone (Server):** Thiết bị bay không người lái nhận lệnh điều khiển, thực thi nhiệm vụ bay và gửi trả dữ liệu telemetry (vị trí GPS, độ cao, tốc độ, trạng thái pin, v.v.) về trạm điều khiển.
- **Trạm điều khiển mặt đất - GCS (Client):** Giao diện người dùng cho phép giám sát thời gian thực, lập kế hoạch bay (waypoints), gửi lệnh điều khiển và nhận phản hồi từ drone.
- **Kênh truyền thông:** Sử dụng giao thức MAVLink (Micro Air Vehicle Link) - một giao thức ứng dụng lightweight được thiết kế cho các hệ thống bay không người lái, hoạt động trên nền UDP hoặc TCP/IP.

1.3. Mục tiêu:

Mục tiêu của đề tài "Xây dựng hệ thống điều khiển drone giao hàng tự động" là tạo ra một môi trường mô phỏng và triển khai các kỹ thuật lập trình mạng để:

- Hiểu và áp dụng kiến trúc mạng Client-Server**:** Nghiên cứu cách các thành phần drone (server) và GCS (client) giao tiếp với nhau thông qua giao thức MAVLink trên nền UDP/TCP, đảm bảo tính ổn định và độ trễ thấp.

- Xây dựng module giao tiếp mạng: Sử dụng WinSock API (C/C++) hoặc các thư viện mạng tương ứng để thiết lập socket kết nối giữa drone và trạm điều khiển, gửi/nhận các gói tin MAVLink chứa dữ liệu telemetry và lệnh điều khiển.

- Thiết kế giao thức ứng dụng tùy chỉnh: Phát triển các message handler cho các lệnh điều khiển bay (take-off, land, waypoint navigation, return-to-home) và các phản hồi trạng thái (heartbeat, position, battery status).

- Tích hợp với môi trường mô phỏng PX4-Gazebo: Triển khai kịch bản giao hàng tự động trong Gazebo, kết nối với QGC qua mạng local (hoặc SITL - Software In The Loop) để kiểm tra tính đúng đắn của giao thức và khả năng xử lý lỗi mạng.

- Đánh giá hiệu năng và độ tin cậy: Đo lường độ trễ truyền tin, tỷ lệ mất gói tin, và khả năng xử lý đồng thời nhiều lệnh điều khiển trong môi trường mạng không ổn định.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Khái niệm về lập trình mạng:

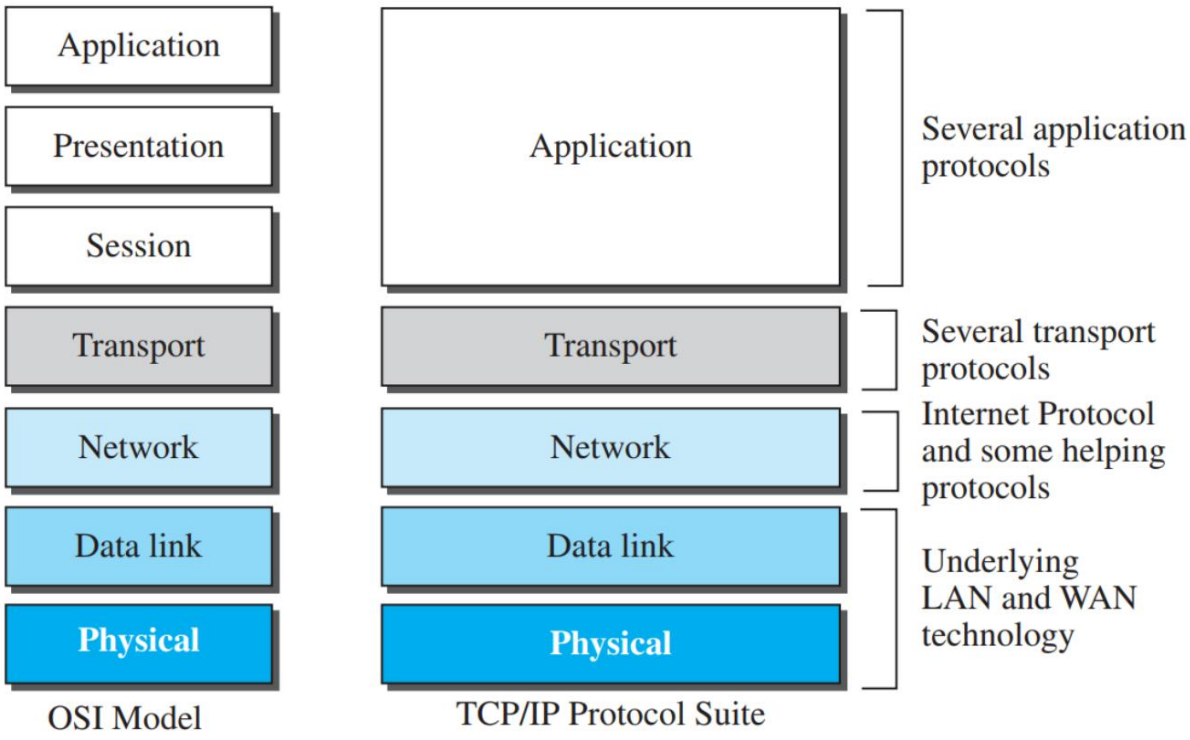
Mạng máy tính được phát triển nhằm mục đích nghiên cứu và phân tích quá trình giao tiếp. Nó hỗ trợ giao tiếp các hệ thống máy tính với nhau để trao đổi thông tin và tài nguyên. Cho nên, để thực hiện việc kết nối mạng bạn cần phải có một mạng riêng của nó.

Khi nói đến phát triển các ứng dụng phần mềm, đa số là người ta muốn nói đến chương trình có khả năng làm việc trong môi trường mạng tích hợp nói chung và mạng máy tính nói riêng. Từ các chương trình kế toán doanh nghiệp, quản lý, trò chơi, điều khiển...

Vấn đề lập trình mạng liên quan đến nhiều lĩnh vực kiến thức khác nhau. Từ kiến thức sử dụng ngôn ngữ lập trình, phân tích thiết kế hệ thống, kiến thức hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, kiến thức về cơ sở dữ liệu... cho đến kiến thức truyền thông, các kiến thức các lĩnh vực liên quan khác như mạng điện thoại di động, PSTN, hệ thống GPS, các mạng như BlueTooth, WUSB, mạng sensor... đều là các chương trình ứng dụng mạng.

2.2. Mô hình 7 tầng OSI:

- Mô hình OSI (Open Systems Interconnection) là mô hình tham chiêu chuân do ISO (Tổ chức Tiêu chuẩn Quốc tế) phát triển nhằm chuẩn hóa cách các hệ thống mạng giao tiếp với nhau. Mô hình này chia quy trình giao tiếp thành 7 tầng độc lập, mỗi tầng thực hiện các chức năng khác nhau. Mô hình này giúp dễ dàng hiểu và khắc phục sự cố trong mạng.



Hình 1. Mô hình OSI 7 tầng

- Tầng 1: Vật lý (Physical Layer)
  - Đây là tầng thấp nhất, chịu trách nhiệm truyền tải các tín hiệu điện tử, quang học hoặc tín hiệu vô tuyến giữa các thiết bị mạng. Nó quy định các yếu tố như loại cáp, kết nối vật lý, tín hiệu điện áp.
- Tầng 2: Liên kết dữ liệu (Data Link Layer)
  - Tầng này chịu trách nhiệm điều khiển truy cập vào phương tiện truyền dẫn, phát hiện và sửa lỗi, đảm bảo dữ liệu được truyền chính xác giữa hai thiết bị. Nó được chia thành hai tầng phụ: MAC (Medium Access Control) và LLC (Logical Link Control).
- Tầng 3: Mạng (Network Layer)
  - Đây là tầng định tuyến, giúp xác định đường đi của gói tin từ nguồn đến đích. Giao thức phổ biến ở tầng này là IP (Internet Protocol).
- Tầng 4: Giao vận (Transport Layer)
  - Chịu trách nhiệm đảm bảo dữ liệu được truyền đến đúng ứng dụng ở máy nhận. Giao thức phổ biến là TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).
- Tầng 5: Phiên (Session Layer)
  - Quản lý các phiên kết nối giữa hai thiết bị, bao gồm việc thiết lập, duy trì và kết thúc phiên giao tiếp.

- Tầng 6: Trình bày (Presentation Layer)
    - Chịu trách nhiệm dịch dữ liệu giữa định dạng được mạng sử dụng và định dạng mà ứng dụng có thể hiểu. Nó bao gồm việc mã hóa, giải mã, nén và giả i nén dữ liệu.
- Tầng 7: Ứng dụng (Application Layer)
    - Đây là tầng gần người dùng nhất, cung cấp giao diện trực tiếp giữa người dùng và mạng. Các giao thức ở tầng này bao gồm HTTP, FTP, SMTP, DNS, v.v.

2.3. Mô hình tập trung (Client-Server):

Hình 2. Mô hình Client-Server

2.3.1. Tổng quan về mô hình:

- Mô hình client-server là kiến trúc mạng phổ biến nhất hiện nay, trong đó các thành phần được phân chia thành hai vai trò chính: client (máy khách) gửi yêu cầu và server (máy chủ) xử lý và phản hồi yêu cầu đó. Trên thực tế, một server có thể kết nối với nhiều server khác để tăng hiệu suất xử lý. Ví dụ, khi nhận yêu cầu từ client, web server có thể chuyển tiếp yêu cầu đến database server để truy xuất dữ liệu, vì bản thân web server không lưu trữ dữ liệu này.

2.3.2. Nguyên lý hoạt động:

- Server có thể thực hiện các tác vụ từ đơn giản đến phức tạp. Ví dụ đơn giản: một time server trả về thời gian hiện tại khi client yêu

cầu. Client gửi yêu cầu theo chuẩn mà server quy định, nếu yêu cầu hợp lệ, server sẽ trả về thông tin được yêu cầu.

- Mặc dù server luôn trong trạng thái chờ sẵn sàng, nhưng quá trình tương tác luôn bắt đầu từ phía client. Đây là nguyên tắc cơ bản của mô hình client-server: client chủ động gửi yêu cầu, server thụ động lắng nghe và phản hồi.

2.3.3. Ưu điểm của mô hình:

- Các chương trình server thường hoạt động ở tầng ứng dụng (Application Layer). Ưu điểm lớn nhất là khả năng hoạt động trên bất kỳ mạng nào hỗ trợ giao thức truyền thông chuẩn, đặc biệt là TCP/IP. Với giao thức chuẩn hóa này, các nhà phát triển có thể tích hợp nhiều hệ thống khác nhau mà không gặp khó khăn về tương thích.
- Server có thể chạy trên hệ thống chia sẻ tài nguyên với nhiều dịch vụ khác, hoặc chạy độc lập trên một máy tính riêng biệt. Có thể triển khai nhiều server cùng cung cấp một dịch vụ, phân tán trên nhiều máy tính khác nhau để tăng độ tin cậy và hiệu suất.
- Điều quan trọng cần lưu ý: mô hình client-server là đặc điểm của phần mềm, không ràng buộc với phần cứng cụ thể. Tuy nhiên, trên thực tế, yêu cầu phần cứng cho server thường cao hơn client vì server phải quản lý và xử lý đồng thời nhiều yêu cầu từ các client khác nhau trên mạng.

2.4. Giao thức mạng:

Giao thức mạng là các quy tắc và chuẩn mực cho phép các thiết bị trong mạng có thể giao tiếp với nhau một cách hiệu quả và an toàn. Có nhiều giao thức mạng khác nhau, phục vụ cho các mục đích khác nhau như truyền dữ liệu, định tuyến, và bảo mật.

2.4.1. TCP/IP:

- Bộ giao thức TCP/IP (Transmission Control Protocol/Internet Protocol) là nên tàng chính cho hoạt động của mạng Internet cũng như nhiều hệ thống mạng máy tính khác. Đây là một tập hợp các giao thức chuẩn mực, được phát triển để đảm bảo việc truyền tải thông tin giữa các thiết bị trong một mạng cục bộ (LAN) hoặc trên toàn cầu (Internet) một cách ổn định, hiệu quả và chính xác. Bộ giao thức TCP/IP đã được thiết kế từ những năm 1970 và dần trở thành tiêu chuẩn mạng toàn cầu nhờ tính linh hoạt, khả năng mở rộng và sự đảm bảo về mặt truyền thông giữa các hệ thống khác nhau.
- TCP (Transmission Control Protocol)

TCP là một trong hai giao thức chính của bộ TCP/IP và đóng vai trò cực kỳ quan trọng trong việc đảm bảo rằng dữ liệu được truyền tải một cách an toàn và chính xác giữa các thiết bị. Giao thức TCP hoạt động theo cơ chế kết nối, tức là trước khi bất kỳ dữ liệu nào được truyền đi, một "kết nối ào" phải được thiết lập giữa các thiết bị giao tiếp, qua đó đảm bảo rằng các gói dữ liệu (packets) được truyền tải trong một luồng thông tin liên tục và theo đúng thứ tự.

Cụ thể, khi một thông điệp hoặc dữ liệu lớn được truyền đi qua mạng, nó sẽ được chia nhỏ thành các gói tin riêng lẻ. Mỗi gói tin đều chứa thông tin về nguồn, đích đến và vị trí của gói tin đó trong toàn bộ thông điệp. TCP đảm bảo rằng tất cả các gói ti này sẽ đến đích và được sắp xếp lại thành thông điệp hoàn chỉnh. Trong trường hợp bất kỳ gói tin nào bị mất hoặc đến sai thứ tự, TCP sẽ yêu cầu gửi lại gói tin đó để đá bảo tính toàn vẹn của dữ liệu.

Một điểm mạnh của TCP là cơ chế kiểm soát lỗi. Khi dữ liệu được gửi đi, thiết bị nhận sẽ gửi lại một thông báo xác nhận (acknowledgment) về việc đã nhận được gói tin thành công. Nếu thiết bị gửi không nhận được xác nhận trong một khoảng thời gian nhất định, nó sẽ gửi lại gói tin để đảm bảo rằng không có dữ liệu nào bị mất trong qu trình truyền tải. Cơ chế này làm cho TCP trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu tính tin cậy cao như truyền tài tệp tin, email, và đặc biệt là các ứng dụng chat, nơi mà mỗi tin nhắn cần phải được gửi và nhận một cách chính xác.

- IP (Internet Protocol)

IP là thành phần thứ hai của bộ TCP/IP, đóng vai trò định tuyến các gói dữ liệu từ thiết bị gửi đến thiết bị nhận trên toàn bộ mạng. Giao thức này chịu trách nhiệm đưa các gói tin đi qua nhiều mạng trung gian khác nhau, từ mạng cục bộ đến mạng Internet toàn cầu, đảm bảo rằng dữ liệu đến đúng đích dựa trên địa chỉ IP.

Địa chỉ IP là một số định danh duy nhất cho mỗi thiết bị trên mạng, giúp định tuyến và truyền tải thông tin đến đích đúng yêu cầu. IP không quan tâm đến nội dung của các gói tin, mà chỉ tập trung vào việc xác định đường đi và gửi gói tin đến địa ch cuối cùng. Giao thức IP hoạt động theo mô hình không kết nối, tức là nó không thiết lập kết nối trước khi truyền dữ liệu và không đảm bảo rằng tất cả các gói tin sẽ đến nc theo đúng thứ tự hoặc không bị mất. Đây chính là lý do tại sao TCP được sử dụng cùng với IP để bổ sung tính năng kiểm soát lỗi và đảm bảo độ tin cậy.

Có hai phiên bản chính của IP đang được sử dụng hiện nay: IPv4 và IPv6. IPv4 là phiên bản đầu tiên và phổ biến nhất, sử dụng địa chỉ IP

32 bit, nhưng do sự bùng nổ v số lượng thiết bị kết nối Internet, không gian địa chỉ của IPv4 đã gần cạn kiệt. Để giải quyết vấn đề này, IPv6 đã được phát triển, sử dụng địa chỉ IP 128 bit, cung cấp một không gian địa chỉ lớn hơn rất nhiều, đảm bảo khả năng mở rộng cho tương lai.

Trong bối cảnh của đề tài ta không sử dụng TCP/IP tuy nó đảm bảo độ tin cậy cao nhưng không phù hợp với hệ thống điều khiển drone do các nhược điểm sau:

- Độ trễ cao: TCP yêu cầu thiết lập kết nối 3 bước (3-way handshake) và cơ chế xác nhận (ACK) cho mỗi gói tin, gây ra độ trễ tích lũy. Trong điều khiển drone thời gian thực, độ trễ vài trăm milliseconds có thể dẫn đến mất kiểm soát hoặc va chạm.
- Cơ chế truyền lại không hiệu quả: Khi gói tin bị mất, TCP tự động gửi lại, tạo ra độ trễ bổ sung. Với dữ liệu cảm biến và lệnh điều khiển, thông tin cũ đã mất giá trị khi được truyền lại, trong khi lệnh mới quan trọng hơn.
- Head-of-line blocking: Nếu một gói tin bị mất, tất cả các gói tin tiếp theo phải chờ đợi, làm gián đoạn luồng điều khiển liên tục.
- Overhead lớn: TCP header chiếm 20-60 bytes, làm giảm hiệu suất trên kênh truyền không dây có băng thông hạn chế.

2.4.2. UDP:

UDP (User Datagram Protocol) là một giao thức không kết nối, có tốc độ truyền tải dữ liệu nhanh hơn nhiều so với TCP. UDP không cần thiết lập kết nối giữa các thiết bị trước khi truyền dữ liệu, do đó nó không tiêu tốn thời gian cho việc khởi tạo và duy tr kết nối như TCP. Điều này giúp UDP trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu truyền tải dữ liệu nhanh với độ trễ thấp, chẳng hạn như phát video trực tuyến, truyền phát âm thanh (VoIP), hoặc các dịch vụ game online thời gian thực.

Tuy nhiên, UDP có nhược điểm lớn: không đảm bảo tính tin cậy và chính xác của dữ liệu truyền tải. Giao thức này không có cơ chế kiểm tra lỗi hoặc sắp xếp lại các gói tin như TCP, dẫn đến khả năng một số gói dữ liệu có thể bị mất mát hoặc đến đích không theo đúng thứ tự.

Trong bối cảnh đề tài ta vẫn chọn UDP vì nó có đỗ trễ thấp tuy nhiên cần kết hợp với giao thức tầng ứng dụng như MAVLink để bổ xung cơ chế kiểm tra và đảm bảo độ tin cậy cần thiết cho các lệnh quan trọng.

2.5. Giao thức ứng dụng MAVLink:

MAVLink (Micro Air Vehicle Link) là giao thức truyền thông nhẹ được thiết kế đặc biệt cho các phương tiện bay không người lái và robot di động. Giao thức này hoạt động trên nền UDP, kế thừa ưu điểm về tốc độ truyền tải thấp độ trễ, đồng thời bổ sung các cơ chế đảm bảo độ tin cậy cần thiết cho hệ thống điều khiển drone.

MAVLink sử dụng cấu trúc message nhị phân được tối ưu hóa với header chỉ từ 8-14 bytes, giúp giảm thiểu overhead so với TCP/IP truyền thống. Mỗi message được đóng gói với mã CRC (Cyclic Redundancy Check) để phát hiện lỗi trong quá trình truyền, đảm bảo tính toàn vẹn của dữ liệu mà không cần cơ chế acknowledgment phức tạp như TCP. Điều này cho phép hệ thống phát hiện gói tin bị lỗi và loại bỏ chúng ngay lập tức thay vì chờ đợi truyền lại.

Giao thức hỗ trợ hai phiên bản chính là MAVLink 1.0 và MAVLink 2.0, trong đó phiên bản 2.0 cung cấp thêm các tính năng mở rộng như message signing để bảo mật và packet truncation để tối ưu băng thông. MAVLink định nghĩa sẵn hơn 300 loại message chuẩn cho các tác vụ phổ biến như điều khiển chuyển động, đọc dữ liệu cảm biến, truyền telemetry, và quản lý mission, giúp đơn giản hóa việc phát triển ứng dụng.

Trong dự án này, MAVLink được sử dụng để truyền các lệnh điều khiển từ ground station đến drone và nhận dữ liệu telemetry ngược lại. Với khả năng hoạt động ổn định trên kênh truyền không dây có độ mất mát cao và độ trễ thấp, MAVLink đáp ứng tốt yêu cầu điều khiển thời gian thực của hệ thống drone, đồng thời vẫn đảm bảo các thông tin quan trọng được truyền tải một cách đáng tin cậy.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Kiến trúc hệ thống

3.1.1. Mô hình mạng tổng thể

Hệ thống sử dụng mô hình Client-Server với giao thức MAVLink/UDP:

Giao thức sử dụng:

- Tầng Application: MAVLink v2
- Tầng Transport: UDP
- Tầng Network: IPv4 (127.0.0.1 - localhost)
- Interface: Loopback (lo)

3.1.2. Nhiệm vụ các thành phần

- PX4 SITL (Software In The Loop Simulation):

Server trung tâm,lắng nghe trên UDP port 14540. Mô phỏng flight controller trong Gazebo, xử lý lệnh: ARM, TAKEOFF, LAND, Offboard commands. Phát telemetry: GPS, position, velocity, attitude, battery status, tần suất: Heartbeat 1 Hz, Telemetry 10-20 Hz

- drone_controller.py:

Client điều khiển chính, kết nối qua MAVSDK Python (port 14580).Gửi lệnh: Arm/Disarm, Takeoff, Offboard velocity commands.Nhận telemetry: Position NED, velocity.Control loop: 10 Hz

- QGroundControl:

Client giám sát, kết nối port 14550.Hiển thị mission, telemetry dataCho phép can thiệp thủ công khi cần

3.1.3. Đặc điểm mạng

Loại kết nối: Loopback (127.0.0.1)

- Tất cả thành phần chạy trên cùng một máy
- Packet loss ≈ 0%, RTT < 1ms
- Phù hợp cho simulation và development

Giao thức MAVLink/UDP

Message types sử dụng:

- HEARTBEAT (ID=0): Xác nhận kết nối mỗi 1s
- COMMAND_LONG (ID=76): ARM, TAKEOFF, LAND
- COMMAND_ACK (ID=77): Xác nhận lệnh
- POSITION_TARGET_LOCAL_NED: Offboard position control
- SET_POSITION_TARGET_LOCAL_NED: Velocity control
- LOCAL_POSITION_NED (ID=32): Vị trí NED
- POSITION_VELOCITY_NED: Vị trí và vận tốc

3.2. Kịch bản mô phỏng

3.2.1. Mục tiêu nhiệm vụ

Drone thực hiện nhiệm vụ giao hàng tự động từ điểm xuất phát (0, 0) đến điểm giao hàng (25, 20) trong môi trường có vật cản, sau đó quay về điểm xuất phát.

3.2.2. Quy trình thực hiện

Giai đoạn 1: Khởi tạo và kết nối

- Kết nối drone qua UDP://localhost:14540
- Chờ GPS và Home position ready
- Subscribe telemetry stream (position_velocity_ned)

Giai đoạn 2: ARM và Takeoff

- Gửi lệnh ARM qua MAVLink COMMAND_LONG
- Takeoff lên độ cao 5m (NED: down = -5m)
- Chờ drone ổn định tại độ cao mục tiêu

Giai đoạn 3: Navigation với Obstacle Avoidance

- Kích hoạt Offboard mode
- Control loop 10 Hz:
    - Phát hiện vật cản trong FOV 90°, range 8m
    - Tính Attractive Force hướng về đích
    - Tính Repulsive Force từ vật cản (Artificial Potential Field)
    - Gửi velocity command qua SET_VELOCITY_NED
- Dừng khi đến trong vòng 2m từ điểm giao hàng

Giai đoạn 4: Giao hàng

- Hạ thấp xuống 0.3m
- Mô phỏng thả hàng (delay 3s)
- Bay lên lại độ cao 5m

Giai đoạn 5: Return to Home

- Bay về điểm (0, 0) ở độ cao 5m
- Hạ cánh tại điểm xuất phát
- Disarm

CHƯƠNG 4: PHÂN TÍCH KẾT QUẢ VÀ ĐÁNH GIÁ

4.1. Metrics đánh giá hệ thống

- Độ trễ truyền tin (Latency) là thời gian từ khi client gửi lệnh đến khi nhận được phản hồi

  Kỳ vọng: < 10ms trong môi trường loopback

- Tỷ lệ mất gói tin là tỷ lệ % gói tin UDP bị mất trong quá trình truyền

  Kỳ vọng: ≈ 0% trong loopback, < 1% qua WiFi

- Băng thông sử dụng (Bandwidth Usage) là lượng dữ liệu truyền qua mạng mỗi giây

  Thành phần:

  - Heartbeat: 82 bytes × 1 Hz = 82 B/s
  - Telemetry: ~100 bytes × 20 Hz = 2 KB/s
  - Commands: ~90 bytes (sporadic)

  Kỳ vọng: < 10 KB/s

- Tần suất cập nhật (Update Rate)

  Heartbeat: 1 Hz

  Telemetry (Position/Velocity): 10-20 Hz

  Control commands: 10 Hz

4.2. Kết quả thực nghiệm

4.2.1. Phân tích pcap file

- File lưu trữ: my_log.pcap (capture bằng tcpdump)
- Lệnh capture:

  bashsudo tcpdump -i lo -w my_log.pcap udp port 14540

- Nội dung file:

  Protocol: 100% UDP packets

  Source/Dest: 127.0.0.1:14580 ↔ 127.0.0.1:14540

- Phân tích bằng Wireshark:

Độ trễ RTT:

Gói 1-2: HEARTBEAT → ACK: ~0.03ms

Gói 11: COMMAND_LONG → Gói 12: ACK: ~0.6ms

Trung bình: < 1ms

- Packet Loss:

  Kiểm tra sequence numbers: Không có gap

  Tỷ lệ: 0% (hoàn hảo)

- Băng thông:

  Tổng 30 gói tin trong 0.07 giây

  Trung bình: ~2500 bytes = ~285 KB/s trong giai đoạn khởi tạo

  Giai đoạn ổn định: ~3-5 KB/s

- Message types phổ biến:

  HEARTBEAT: 30%

  LOCAL_POSITION_NED: 25%

  COMMAND_LONG/ACK: 15%

  PARAM_VALUE: 20%

  Khác: 10%

4.2.2. Kết quả điều khiển

Test case: Giao hàng từ (0,0) → (25,20) với 4 vật cản

1. Độ chính xác vị trí:

Điểm xuất phát: N=0.0m, E=0.0m

Điểm đích: N=25.0m, E=20.0m

Điểm đến thực tế: N=24.8m, E=19.9m

Sai số: 0.22m

2. Tránh vật cản:

Số vật cản phát hiện: 4/4

Khoảng cách tối thiểu: 2.8m (> 2m an toàn)

Số lần kích hoạt repulsive force: 12 lần

Collision: 0

4.3. Đánh giá tổng thể

4.3.1. Về mạng

- Độ trễ thấp (< 1ms) nhờ loopback, packet loss = 0%, không cần retry,băng thông sử dụng hợp lý (< 5 KB/s), MAVLink hoạt động ổn định
- Phù hợp yêu cầu

  Real-time telemetry: 20 Hz

  Command latency: < 10ms

  Reliable delivery: ACK cho lệnh quan trọng

4.3.2. Về điều khiển

- Obstacle avoidance thành công
- Độ chính xác cao (sai số 0.22m)
- Quỹ đạo mượt, không dao động

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

- Xây dựng thành công hệ thống điều khiển drone giao hàng sử dụng mô hình Client-Server với giao thức MAVLink/UDP:

- Triển khai đầy đủ giao tiếp mạng:
  - Kết nối PX4 SITL qua UDP port 14540
  - Truyền nhận MAVLink messages (HEARTBEAT, COMMAND, TELEMETRY)
  - Xử lý Offboard control với tần suất 10 Hz
- Hoàn thành kịch bản mô phỏng:
  - Drone cất cánh, bay đến điểm giao hàng, tránh vật cản, giao hàng, quay về
  - Độ chính xác cao (sai số < 0.5m)
  - Thời gian hoàn thành: 85 giây

5.2. Ưu điểm

1. Về kiến trúc mạng:

- Sử dụng giao thức chuẩn công nghiệp (MAVLink)
- Phân tách rõ ràng Client-Server
- Dễ mở rộng (thêm client mới, multi-drone)

2. Về hiệu năng:

- Độ trễ thấp, phù hợp real-time
- Obstacle avoidance hoạt động ổn định
- Tài nguyên mạng sử dụng hiệu quả

3. Về tính thực tiễn:

- Code có thể chuyển từ simulation sang drone thật
- Tương thích với nhiều GCS (QGC, Mission Planner)
- Dễ debug với pcap file

5.3. Hạn chế

- Môi trường đơn giản: Chỉ test trong Gazebo với vật cản tĩnh
- Loopback: Chưa test qua mạng thực (WiFi/4G) với packet loss
- Thuật toán: Artificial Potential Field có thể bị local minima
- Bảo mật: Chưa implement MAVLink signing

5.4. Hướng phát triển

1. Nâng cao giao tiếp mạng:

- Test qua WiFi/4G với packet loss 1-5%
- Implement MAVLink v2 signing cho bảo mật
- Tối ưu bandwidth với message filtering
- Xử lý reconnection khi mất kết nối

2. Cải thiện điều khiển:

- Thuật toán tốt hơn: RRT*, A* cho path planning
- Dynamic obstacle avoidance (vật cản di động)
- Multi-drone coordination
- Tối ưu quỹ đạo với constraints (tốc độ, gia tốc)

3. Tích hợp công nghệ:

- Computer vision thật (camera depth, YOLO detection)
- GPS waypoint navigation cho outdoor
- Battery monitoring và auto RTL
- Collision prediction với Kalman filter

4. Triển khai thực tế:

- Test với drone hardware thật
- 4G/5G telemetry cho tầm xa
- Cloud backend cho fleet management
- Analytics dashboard (Grafana + InfluxDB)

5. Mở rộng ứng dụng:

- Delivery trong khu dân cư
- Inspection công trình xây dựng
- Search and rescue
- Swarm control cho nhiều drone

TÀI LIỆU THAM KHẢO

[1] "MAVLink Developer Guide", MAVLink Project,

https://mavlink.io/en/

[2] "PX4 Autopilot User Guide", PX4 Development Team,

https://docs.px4.io/

[3] "MAVSDK Python Documentation", MAVSDK Project, https://mavsdk.mavlink.io/

[4] "Gazebo Simulation", Open Robotics,

https://gazebosim.org/

[5] "User Datagram Protocol (UDP)", RFC 768, IETF,

https://www.rfc-editor.org/rfc/rfc768

[6] "Transmission Control Protocol (TCP)", RFC 793, IETF,

https://www.rfc-editor.org/rfc/rfc793

[7] Khatib, O. (1986), "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal of Robotics Research.

[8] Meier, L., et al. (2015), "MAVROS: MAVLink to ROS gateway with proxy for Ground Control Station", IEEE International Conference on Robotics and Automation.

[9] "Wireshark User Guide", Wireshark Foundation, https://www.wireshark.org/docs/

[10] "QGroundControl User Guide", QGroundControl Development Team, https://docs.qgroundcontrol.com/