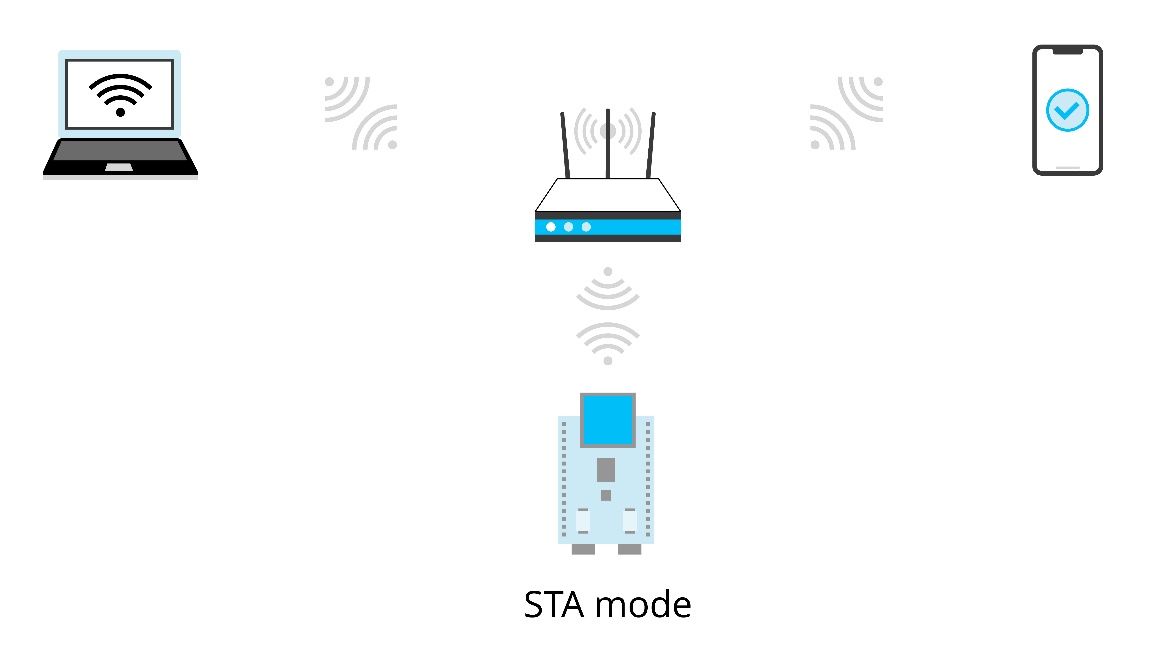
# Chương 1: Kiểm soát cục bộ ESP32 (Local Control)

## Giới thiệu

### 1.1 Khái niệm

Kiểm soát cục bộ (Local Control) là khái niệm điều khiển các thiết bị lân cận trong một khoảng cách nhất định. ESP32 cung cấp hai công nghệ kiểm soát cục bộ thường được sử dụng là thông qua Bluetooth hoặc Wi-Fi trong mạng LAN:

* *Kiểm soát cục bộ qua Wi-Fi:* Thiết bị gửi lệnh và thiết bị nhận lệnh cần được kết nối trong cùng 1 mạng.
* *Kiểm soát cục bộ qua Bluetooth:* Có thể gửi trực tiếp dữ liệu từ Smartphone sang thiết bị.



Hình 1: Cấu trúc mạng cục bộ bằng Wi-Fi trong mạng LAN

Chọn sử dụng Wi-Fi hay Bluetooth phụ thuộc vào việc thiết bị có cần kết nối mạng không:

- *Nếu không:* Nên sử dụng Bluetooth vì Bluetooth dễ dàng cấu hình hơn, không cần router.

- *Nếu có:* Nên sử dụng Wi-Fi vì Wi-Fi hỗ trợ đa dạng các thiết bị, có thể nhận lệnh điều khiển từ nhiều thiết bị cùng lúc. Ngoài ra, băng thông của Wi-Fi cũng lớn hơn Bluetooth.

### 1.2 Ứng dụng

Hầu hết các thiết bị IoT được kết nối với nền tảng đám mây để chuyển tiếp lệnh từ điện thoại thông minh để thực hiện điều khiển từ xa. Việc kiểm soát như vậy phụ thuộc vào Internet do bộ định tuyến Wi-Fi cung cấp để duy trì liên kết giữa thiết bị được kiểm soát và nền tảng đám mây. Nhưng một khi bộ định tuyến Wi-Fi ngắt kết nối Internet, điều khiển từ xa sẽ bị tê liệt. Tại thời điểm này, kiểm soát cục bộ (Local Control) sẽ là sự bổ sung tốt cho việc gửi lệnh, do đó ngăn ngừa nguy cơ thiết bị không thể điều khiển được do không có kết nối mạng.

Khung điều khiển cục bộ dựa trên Wi-Fi trong mạng LAN bao gồm bộ định tuyến Wi-Fi, thiết bị điều khiển và thiết bị được điều khiển. Thiết bị điều khiển có thể là điện thoại thông minh hoặc máy tính có thể chạy ngăn xếp giao thức TCP/IP. Nó phải được kết nối với cùng một bộ định tuyến Wi-Fi với thiết bị được điều khiển để đảm bảo rằng chúng nằm trong cùng một mạng LAN để truyền dữ liệu.

Đối với điều khiển cục bộ dựa trên Bluetooth, không cần bộ định tuyến Wi-Fi. Điện thoại thông minh có thể kết nối trực tiếp với thiết bị được điều khiển thông qua Bluetooth và thực hiện truyền dữ liệu điểm-điểm.

### 1.3 Ưu điểm

***Phản hồi nhanh:*** Kiểm soát cục bộ chỉ yêu cầu dữ liệu được truyền trong mạng LAN thay vì qua Internet. Do đó, nó hoạt động với độ trễ ngắn hơn và phản hồi nhanh hơn.

***Bảo mật:*** Vì dữ liệu chỉ tương tác trong mạng LAN bao gồm thiết bị được điều khiển, bộ định tuyến Wi-Fi và thiết bị điều khiển nên khả năng dữ liệu bị đánh cắp hoặc sửa đổi sẽ thấp hơn, do đó tăng cường quyền riêng tư và bảo mật dữ liệu.

***Tiết kiệm băng thông:*** Kiểm soát cục bộ giúp tiết kiệm băng thông Internet. Nó không dễ bị ngắt kết nối, miễn là hai bên ở trong cùng một mạng LAN hoặc được liên kết qua Bluetooth thì việc kiểm soát có thể được duy trì. Đối với một số sản phẩm không có cần truy cập vào nền tảng đám mây, kiểm soát cục bộ là cách duy nhất để kiểm soát qua điện thoại thông minh.

### 1.4 Điều khiển cục bộ qua Wi-Fi

Đối với điều khiển cục bộ dựa trên phương tiện truyền dẫn không dây Wi-Fi, dữ liệu sẽ chạy trên ngăn xếp giao thức TCP/IP. Do tất cả dữ liệu đều được truyền dựa trên lớp IP nên biết địa chỉ IP là điều kiện tiên quyết cho quá trình truyền dữ liệu tiếp theo. Sau khi lấy địa chỉ IP của thiết bị được điều khiển, điện thoại thông minh có thể giao tiếp với thiết bị được điều khiển thông qua giao thức TCP/IP hoặc giao thức UDP.

Với tư cách là người nhận, thiết bị được điều khiển sẽ nhận các lệnh điều khiển do điện thoại thông minh gửi trong điều khiển cục bộ; và điện thoại thông minh, với tư cách là người gửi, sẽ gửi lệnh điều khiển đến thiết bị được điều khiển. Vì vậy, thiết bị được điều khiển đóng vai trò là máy chủ; và điện thoại thông minh đóng vai trò là máy khách, cho phép nhiều máy khách gửi lệnh điều khiển đến máy chủ.

### 1.5 Điều khiển cục bộ qua Bluetooth

Đối với các khung điều khiển cục bộ dựa trên điều khiển Bluetooth, Bluetooth của thiết bị được điều khiển sẽ phát thông tin Bluetooth của chính nó và điện thoại thông minh chỉ cần quét Bluetooth của thiết bị được điều khiển. Việc tìm kiếm thiết bị được điều khiển qua Bluetooth đơn giản hơn nhiều so với Wi-Fi. Sau khi điện thoại thông minh kết nối với Bluetooth của thiết bị được điều khiển, nó có thể gửi dữ liệu đến thiết bị. Ngoài ra, việc truyền Bluetooth không phụ thuộc vào ngăn xếp giao thức TPC/IP vì nó có giao thức truyền riêng.

## 2. Phương pháp tìm kiếm các thiết bị được điều khiển

Trong giao thức TCP/IP, việc tìm kiếm các thiết bị được điều khiển có nghĩa là lấy địa chỉ IP của thiết bị được điều khiển.

Khám phá cục bộ ***(Local Discovery)*** là quét thông tin về các nút trong mạng LAN, bao gồm thông tin địa chỉ để liên lạc với các nút, thông tin dịch vụ ứng dụng được các nút hỗ trợ và thông tin tùy chỉnh. Ví dụ: mDNS (DNS Multicast) là giao thức khám phá cục bộ thường được sử dụng. Nguyên tắc khám phá cục bộ là gửi bản tin và thiết bị ngang hàng sẽ thông báo cho thiết bị gửi thông tin thiết bị của mình sau khi nhận được bản tin.

Cần đảm bảo rằng thiết bị ngang hàng có thể nhận được tin nhắn do thiết bị gửi gửi. Trên thực tế, ngoài giao tiếp điểm-điểm (*unicast*) được sử dụng phổ biến hơn, còn có giao tiếp một-nhiều (*multicast*) và một-- tất cả các thông tin liên lạc (*broadcast*). Địa chỉ IP có thể được chia thành địa chỉ unicast, địa chỉ multicast và địa chỉ quảng bá. Unicast cần biết địa chỉ IP của thiết bị ngang hàng nên không phù hợp để khám phá cục bộ. Multicast và Broadcast không cần biết địa chỉ IP của thiết bị ngang hàng. Họ gửi tin nhắn đến các địa chỉ cụ thể và thiết bị ngang hàng có thể nhận tin nhắn miễn là nó giám sát địa chỉ này. Do đó, Multicast và Broadcast phù hợp để khám phá các thiết bị trong mạng LAN và thiết bị ngang hàng có thể nhận được tin nhắn do người gửi gửi bằng hai công nghệ này.

### 2.1 Broadcast

Broadcast đề cập đến việc gửi tin nhắn đến tất cả các máy thu có thể có trong mạng. Có hai ứng dụng chính của Broadcast:

- Định vị một máy chủ trong mạng cục bộ.

- Giảm lưu lượng gói trong mạng cục bộ để một tin nhắn có thể thông báo cho tất cả các máy chủ trong mạng cục bộ.

Các bản tin Broadcast phổ biến bao gồm:

* *Giao thức phân giải địa chỉ (ARP):* Nó có thể được sử dụng để phát yêu cầu ARP trong mạng cục bộ: "Vui lòng cho tôi biết địa chỉ MAC của thiết bị có địa chỉ IP a.b.c.d là gì". Phát sóng ARP là phát sóng MAC trên lớp 2 (lớp liên kết), không phải phát sóng IP trên lớp 3 (lớp mạng).
* *Giao thức cấu hình máy chủ động (DHCP):* Nếu có máy chủ DHCP trên mạng cục bộ, máy khách DHCP sẽ gửi yêu cầu DHCP cho địa chỉ IP đích (thường là 255.255.255.255) và máy chủ DHCP trên cùng mạng có thể nhận yêu cầu và trả lời bằng địa chỉ IP được chỉ định.

Broadcast chủ yếu sử dụng giao thức UDP thay vì giao thức TCP vì nó phù hợp hơn với Unicast.

#### **2.1.1 Địa chỉ Broadcast**

Địa chỉ quảng bá bao gồm địa chỉ quảng bá MAC trên lớp 2 (lớp liên kết) (FF:FF:FF:FF:FF:FF) và địa chỉ quảng bá IP trên lớp 3 (lớp mạng) (255.255.255.255), sau đây gọi là địa chỉ lớp 2 và địa chỉ lớp 3. Phần này chủ yếu giới thiệu địa chỉ lớp 3. Nói chung, khi địa chỉ lớp 3 của tin nhắn đều là 255 thì địa chỉ lớp 2 thường là FF. Điều này là do một bản tin có địa chỉ lớp 3 đầy 255 có nghĩa là tất cả các thiết bị trong mạng cục bộ sẽ nhận được tin nhắn này. Nếu địa chỉ lớp 2 của tin nhắn không phải là toàn bộ FF, tin nhắn sẽ bị loại bỏ trong quá trình xử lý địa chỉ lớp 2 của thiết bị nhận. Đối với thiết bị nhận, nếu địa chỉ lớp 2 của tin nhắn không phải là địa chỉ quảng bá, cũng không phải địa chỉ MAC cục bộ và địa chỉ MAC multicast (chẳng hạn như 01:00:5E:XX:XX:XX), nó sẽ bị loại bỏ và không xử lý. Do đó, nói chung, nếu địa chỉ lớp 3 là địa chỉ quảng bá thì địa chỉ lớp 2 cũng vậy.

Địa chỉ IPv4 bao gồm ID mạng con và ID máy chủ. Ví dụ: đối với một thiết bị có địa chỉ IP là 192.168.3.4 và mặt nạ mạng con là 255.255.255.0, ID mạng con và ID máy chủ của thiết bị đó được tính từ địa chỉ IP và mặt nạ mạng con. Trong ví dụ này, ID mạng con là 192.168.3.0 và ID máy chủ là 4. Khi ID mạng con và ID máy chủ đều là 255 thì đó là địa chỉ quảng bá; nó cũng là địa chỉ quảng bá khi chỉ có ID máy chủ là 255. Ví dụ: nếu mạng con là 192.168.1/24 thì 192.168.1.255 là địa chỉ quảng bá của mạng con này.

Phạm vi quảng bá của địa chỉ đầu tiên lớn hơn địa chỉ của một mạng con cụ thể. Ví dụ: bộ định tuyến Wi-Fi có hai mạng con là 192.168.1/24 và 192.168.2/24. Nếu máy chủ có địa chỉ IP 192.168.1.2 trong mạng con 192.168.1/24 gửi tin nhắn đến địa chỉ đích 192.168.1.255 thì bộ định tuyến Wi-Fi sẽ chỉ chuyển tiếp tin nhắn đến máy chủ trong mạng con 192.168.1/24, và sẽ không chuyển tiếp nó tới máy chủ trong mạng con 192.168.2/24. Nếu máy chủ gửi tin nhắn đến địa chỉ đích 255.255.255.255, bộ định tuyến Wi-Fi sẽ chuyển tiếp tin nhắn đến máy chủ trong cả hai mạng con. Do đó, địa chỉ quảng bá có ID máy chủ là 255 còn được gọi là địa chỉ quảng bá hướng mạng con. Bằng cách sử dụng địa chỉ quảng bá hướng mạng con, bạn có thể gửi tin nhắn đến một mạng con được chỉ định để những tin nhắn này sẽ không được gửi đến các mạng con không cần chúng trong mạng LAN, do đó tiết kiệm tài nguyên mạng.

### 2.2 Multicast

Multicast đề cập đến việc gửi tin nhắn đến người nhận quan tâm. So với unicast và broadcast, multicast cung cấp một giải pháp trung hòa. Multicast chủ yếu nhấn mạnh đến khái niệm nhóm, nghĩa là máy chủ có thể gửi tin nhắn đến địa chỉ nhóm và tất cả các máy chủ tham gia nhóm này đều có thể nhận được tin nhắn. Điều này hơi giống với việc phát sóng theo hướng mạng con, nhưng linh hoạt hơn vì máy chủ có thể tham gia hoặc rời khỏi một nhóm nhất định bất kỳ lúc nào, do đó giảm gánh nặng cho mạng cục bộ và máy chủ.

Giao thức quản lý nhóm Internet (IGMP) là giao thức chịu trách nhiệm quản lý các thành viên IP multicast, được sử dụng để thiết lập và duy trì mối quan hệ thành viên nhóm multicast giữa máy chủ IP và các bộ định tuyến Wi-Fi multicast lân cận trực tiếp của nó. Đối với phát multicast, bộ định tuyến Wi-Fi cần hỗ trợ giao thức IGMP.

#### **2.2.1 Địa chỉ multicast**

Địa chỉ đích của tin nhắn multicast sử dụng địa chỉ IP lớp D. Byte đầu tiên bắt đầu bằng số nhị phân 1110 và có phạm vi từ 224.0.0.0 đến 239.255.255.255. Vì địa chỉ IP multicast xác định một nhóm máy chủ nên địa chỉ IP multicast chỉ có thể được sử dụng làm địa chỉ đích chứ không phải địa chỉ nguồn, địa chỉ nguồn luôn là địa chỉ unicast.

Nhóm multicast là một nhóm được xác định bởi một địa chỉ multicast cụ thể. Khi các thành viên trong hoặc ngoài nhóm gửi tin nhắn đến địa chỉ multicast này, các thành viên nhóm được xác định bởi địa chỉ multicast có thể nhận được tin nhắn. Các nhóm multicast có thể là vĩnh viễn hoặc tạm thời. Trong số các địa chỉ multicast, các địa chỉ multicast được gán chính thức được gọi là nhóm multicast cố định, trong khi những địa chỉ không dành riêng hay cố định được gọi là nhóm multicast tạm thời. Số lượng máy chủ trong nhóm multicast cố định và tạm thời là động và thậm chí có thể bằng 0.

Địa chỉ multicast được phân loại như sau:

- *224.0.0.0 ~ 224.0.0.255:* Địa chỉ multicast dành riêng (nhóm multicast cố định). Địa chỉ 224.0.0.0 không được phân bổ và các địa chỉ khác được sử dụng cho các giao thức định tuyến.

- *224.0.1.0 ~ 224.0.1.255*: Địa chỉ multicast công cộng, có thể được sử dụng trên Internet.

- *224.0.2.0 ~ 238.255.255.255:* Địa chỉ multicast có sẵn cho người dùng (nhóm multicast tạm thời), có giá trị trên toàn mạng.

- *239.0.0.0 ~ 239.255.255.255*: Địa chỉ multicast dành cho quản lý cục bộ, chỉ hợp lệ trong phạm vi cục bộ cụ thể.

### 2.3 So sánh giữa Broadcast và Multicast

|  |  |  |
| --- | --- | --- |
| **So sánh** | **Broadcast** | **Multicast** |
| Nguyên tắc | Bản tin gửi tới tất cả các địa chỉ trong mạng | Bản tin chỉ gửi đến địa chỉ đã quan tâm |
| Truyền | Một – Tất cả | Một – Nhiều |
| Quản lý | Không cần quản lý theo nhóm | Cần quản lý theo nhóm |
| Mạng | Có thể gây tốn băng thông không cần thiết | Có thể kiểm soát lượng băng thông tiêu hao |
| Tốc độ | Chậm | Nhanh |

Bảng 1: So sánh Broadcast và Multicast

### 2.4 Ứng dụng Multicast DNS (mDNS)

Trong mạng máy tính, giao thức Multicast DNS (mDNS) phân giải tên máy chủ thành địa chỉ IP trong các mạng nhỏ không bao gồm máy chủ tên cục bộ. Đây là một máy chủ có cấu hình không. mDNS về cơ bản có giao diện lập trình, định dạng gói và chế độ hoạt động giống như hệ thống tên miền (DNS) truyền thống.

#### 2.4.1 Khái niệm giao thức mDNS

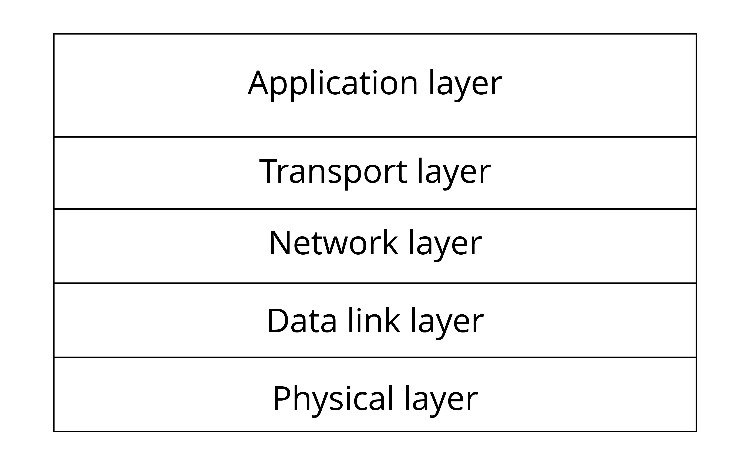
mDNS là giao thức phân giải tên miền dành cho mạng cục bộ, sử dụng cổng 5353 và địa chỉ multicast 224.0.0.251. Nó là một giao thức ứng dụng chạy trên UDP. Không giống như các giao thức DNS truyền thống, mDNS không yêu cầu máy chủ DNS thực hiện phân giải tên miền, điều này giúp tránh rắc rối khi định cấu hình máy chủ tên miền trên mạng cục bộ.

Sau khi máy chủ có dịch vụ mDNS được kích hoạt tham gia mạng LAN, trước tiên nó sẽ multicast một bản tin đến địa chỉ multicast 224.0.0.251 của mạng LAN, *"Tôi là ai? Địa chỉ IP của tôi là gì? Tôi cung cấp những dịch vụ và số cổng nào?"*. Sau khi nhận được tin nhắn, các máy chủ khác có dịch vụ mDNS được kích hoạt trên mạng LAN sẽ ghi lại tin nhắn và phản hồi với câu *"Đó là ai? Địa chỉ IP của nó là gì? Dịch vụ và số cổng mà nó cung cấp là gì?"*. Nếu máy chủ muốn truy vấn tên miền mDNS, trước tiên nó sẽ truy vấn thông tin bộ đệm của chính nó. Nếu không tìm thấy, nó sẽ phát đa hướng một truy vấn tới mạng LAN để hỏi địa chỉ IP, dịch vụ và số cổng của tên miền.

Tên miền mDNS khác với tên miền DNS ở hậu tố ".local".

## 3. Điều khiển thiết bị trong mạng cục bộ

### 3.1 Transmission Control Protocol (TCP)



Hình 2: TCP/IP model

TCP là một trong những giao thức chính trong họ giao thức Internet. Trong mô hình TCP/IP, TCP đóng vai trò là giao thức lớp vận chuyển (transport layer protocol), cung cấp khả năng truyền dữ liệu đáng tin cậy cho các giao thức lớp ứng dụng (application layer protocols) như HTTP, MQTT, FTP, v.v.

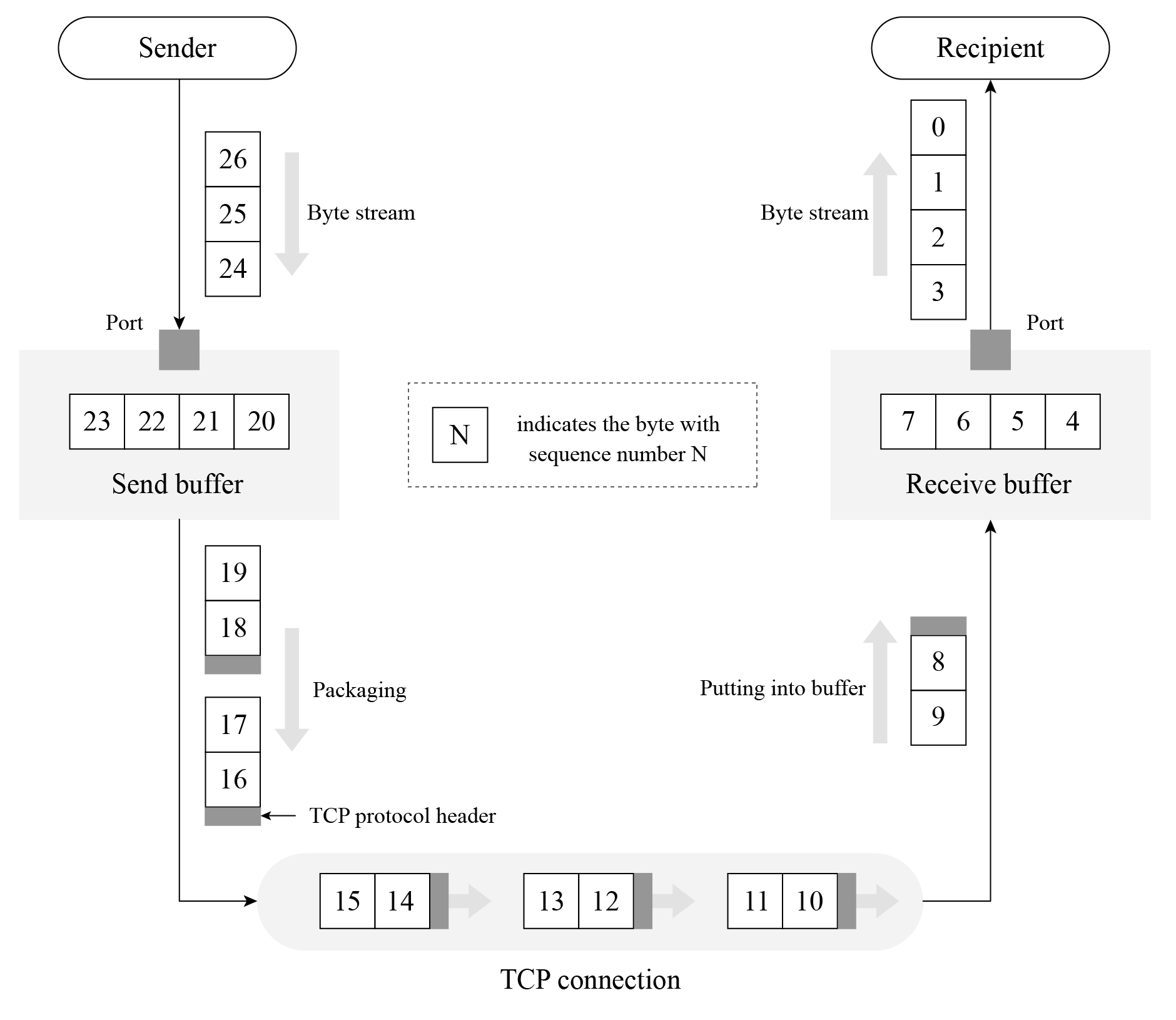
#### 3.1.1 Giới thiệu TCP

TCP là giao thức truyền thông dựa trên luồng byte, định hướng kết nối, đáng tin cậy ở lớp vận chuyển, được xác định bởi RFC 793 của IETF. Đặc điểm của TCP:

* *Định hướng kết nối:* Trước khi gửi dữ liệu bằng TCP, kết nối phải được thiết lập giữa người gửi và người nhận, thường được gọi là bắt tay ba chiều.
* *Đáng tin cậy*: Khi gửi dữ liệu bằng TCP, việc nhận dữ liệu của người nhận có thể được đảm bảo. Nếu dữ liệu bị mất, dữ liệu bị mất sẽ được truyền lại. TCP cũng có thể đảm bảo rằng người nhận nhận dữ liệu theo thứ tự.
* *Dựa trên luồng byte:* Khi gửi dữ liệu bằng TCP, dữ liệu lớp ứng dụng trước tiên được ghi vào bộ đệm TCP. Sau đó, TCP điều khiển việc truyền dữ liệu theo cách dựa trên luồng byte, độc lập với độ dài của thông báo được ghi bởi lớp ứng dụng. Vì vậy, nó là một giao thức dựa trên luồng byte.

Quá trình TCP gửi dữ liệu tới người nhận như sau:

* Chương trình ứng dụng lớp trên (upper-layer application program) ghi dữ liệu vào bộ đệm TCP (Buffer).
* Bộ đệm TCP đóng gói dữ liệu vào một bản tin TCP và gửi nó đến lớp mạng.
* Người nhận nhận được bản tin TCP và đặt nó vào bộ đệm TCP.
* Sau khi nhận được một lượng dữ liệu nhất định, dữ liệu sẽ được sắp xếp và sắp xếp lại trước khi được báo cáo lên lớp ứng dụng.



Hình 3: Truyền nhận dữ liệu sử dụng TCP

### 3.2 HyperText Transfer Protocol (HTTP)

HTTP là một chuẩn cho các yêu cầu (Request) và phản hồi (Response) giữa máy khách (người dùng) và máy chủ (trang web). Máy khách thiết lập kết nối TCP với máy chủ thông qua trình duyệt web, trình thu thập dữ liệu web hoặc các công cụ khác, sau đó gửi yêu cầu đọc dữ liệu máy chủ, tải dữ liệu hoặc biểu mẫu lên máy chủ và đọc trạng thái phản hồi của máy chủ, chẳng hạn như "HTTP /1.1 200 OK", cũng như nội dung được trả về (chẳng hạn như các tệp được yêu cầu, thông báo lỗi hoặc thông tin khác). Các tài nguyên được yêu cầu thông qua HTTP được xác định bằng mã định danh tài nguyên thống nhất (URI).

Trong phiên bản 0.9 và 1.0 của HTTP, kết nối TCP sẽ bị đóng sau mỗi yêu cầu và phản hồi. Trong phiên bản 1.1 của HTTP, một cơ chế duy trì kết nối đã được giới thiệu, cho phép kết nối lặp lại nhiều yêu cầu và phản hồi, giảm thời gian bắt tay TCP và chi phí mạng trước mỗi yêu cầu dữ liệu.

Các phương thức yêu cầu HTTP phổ biến bao gồm:

* GET: Yêu cầu tài nguyên URI được chỉ định.
* POST: Gửi dữ liệu đến tài nguyên URI được chỉ định và yêu cầu máy chủ xử lý dữ liệu đó (chẳng hạn như gửi biểu mẫu hoặc tải tệp lên).
* PUT: ghi đè tất cả thông tin của đối tượng với những gì được gửi lên
* DELETE: Yêu cầu máy chủ xóa tài nguyên được xác định bởi URI.

### 3.3 User Datagram Protocol (UDP)

UDP là một giao thức truyền thông hướng datagram đơn giản, nằm ở lớp vận chuyển như TCP. UDP được thiết kế bởi David P. Reed vào năm 1980 và được định nghĩa trong RFC 768 (trích từ Wikipedia). UDP là một giao thức truyền không đáng tin cậy. Sau khi dữ liệu được gửi qua UDP, lớp bên dưới không giữ lại dữ liệu để tránh mất mát trong quá trình truyền. Bản thân UDP không hỗ trợ sửa lỗi, quản lý hàng đợi hoặc kiểm soát tắc nghẽn nhưng hỗ trợ tổng kiểm tra.

UDP là một giao thức không kết nối. Nó không cần thiết lập kết nối trước khi gửi dữ liệu, không giống như TCP. Dữ liệu có thể được gửi trực tiếp đến thiết bị ngang hàng mà không cần thiết lập kết nối. Vì không cần thiết lập kết nối trong quá trình truyền dữ liệu nên không cần duy trì trạng thái kết nối, bao gồm cả trạng thái gửi và nhận.

UDP chỉ chịu trách nhiệm truyền tải, vì vậy các ứng dụng sử dụng giao thức này cần kiểm soát nhiều hơn cách gửi và xử lý dữ liệu, chẳng hạn như cách đảm bảo rằng các ứng dụng ngang hàng nhận dữ liệu chính xác và theo thứ tự.

So với TCP, UDP không thể đảm bảo việc truyền dữ liệu an toàn và đáng tin cậy. Bạn có thể thắc mắc tại sao giao thức UDP vẫn được sử dụng. Bản chất không kết nối của UDP dẫn đến chi phí mạng và thời gian ít hơn TCP. Việc truyền UDP không đáng tin cậy (chủ yếu là không có khả năng đảm bảo truyền lại sau khi mất gói) phù hợp hơn cho các ứng dụng như phương tiện truyền phát trực tuyến, trò chơi nhiều người chơi trong thời gian thực và giọng nói IP, trong đó việc mất một vài gói sẽ không ảnh hưởng đến ứng dụng. Mặt khác, nếu TCP được sử dụng để truyền lại thì độ trễ của mạng sẽ tăng lên rất nhiều.

### 3.4 Constrained Application Protocol (CoAP)

CoAP là một giao thức tương tự như các ứng dụng web trong thiết bị IoT. Nó được định nghĩa trong RFC 7252 và có thể được sử dụng cho các thiết bị IoT có giới hạn tài nguyên, cho phép các thiết bị có giới hạn tài nguyên đó được gọi là nút giao tiếp với phạm vi Internet rộng hơn bằng các giao thức tương tự. CoAP được thiết kế cho các thiết bị trên cùng một mạng bị hạn chế (chẳng hạn như mạng năng lượng thấp, bị mất dữ liệu), giữa các thiết bị và các nút chung trên Internet cũng như giữa các thiết bị trên các mạng bị hạn chế khác nhau được kết nối bởi Internet.

CoAP dựa trên mô hình yêu cầu và phản hồi, tương tự như HTTP, có thể bù đắp những thiếu sót trong việc truyền UDP không đáng tin cậy và đảm bảo rằng dữ liệu không bị mất hoặc bị rối loạn. Tài nguyên của máy chủ được xác định bằng các URL (chẳng hạn như coap://[IP]/id/light\_status). Máy khách truy cập tài nguyên của máy chủ thông qua URL của tài nguyên và vận hành tài nguyên của máy chủ thông qua bốn phương thức yêu cầu (GET, PUT, POST và DELETE).

CoAP cũng có các tính năng sau:

* Cả máy khách và máy chủ đều có thể gửi yêu cầu cho nhau một cách độc lập.
* Hỗ trợ truyền dữ liệu đáng tin cậy.
* Hỗ trợ phát đa hướng và phát sóng, cho phép truyền dữ liệu một-nhiều.
* Hỗ trợ giao tiếp với mức tiêu thụ điện năng thấp và kết nối không liên tục.
* So với HTTP, tiêu đề của nó nhẹ hơn.

### 3.5 Giao thức Bluetooth

Giao thức Bluetooth xác định các định dạng tin nhắn và quy trình để hoàn thành các chức năng cụ thể, chẳng hạn như kiểm soát liên kết, dịch vụ bảo mật, trao đổi thông tin dịch vụ và truyền dữ liệu. Phần này chỉ giới thiệu giao thức thuộc tính (ATT) của đặc tả giao thức Bluetooth. Dữ liệu Bluetooth tồn tại ở dạng thuộc tính và mỗi thuộc tính bao gồm bốn phần tử.

#### Xử lý thuộc tính

Giống như địa chỉ bộ nhớ được sử dụng để tìm nội dung trong bộ nhớ, các thẻ điều khiển thuộc tính cũng có thể giúp tìm thuộc tính tương ứng. Ví dụ: bộ điều khiển thuộc tính đầu tiên là 0x0001, bộ điều khiển thuộc tính thứ hai là 0x0002, v.v., tối đa là 0xFFFF.

#### Thuộc tính UUID

Mỗi dữ liệu đại diện cho thuộc tính cụ thể. Ví dụ: đèn thông minh có hai thuộc tính cơ bản, một thuộc tính để cài đặt trạng thái bật/tắt và thuộc tính còn lại để đọc trạng thái bật/tắt.

#### Giá trị thuộc tính

Giá trị thuộc tính là thông tin mà mỗi thuộc tính mang theo, trong khi ba phần tử còn lại giúp cho các đối tác ngang hàng có được giá trị thuộc tính dễ dàng hơn nhiều. Ví dụ: đối với đèn thông minh, giá trị thuộc tính để đặt trạng thái bật/tắt có thể được đặt thành "1" để bật đèn hoặc thành "0" để tắt đèn; giá trị thuộc tính để đọc trạng thái bật/tắt có thể là "1" cho trạng thái "bật" hoặc "0" cho trạng thái "tắt".

#### Quyền thuộc tính

Mỗi thuộc tính có các hạn chế truy cập tương ứng đối với các giá trị thuộc tính riêng của nó, chẳng hạn như một số thuộc tính có thể đọc được, một số thuộc tính có thể ghi và một số thuộc tính có thể đọc và ghi được. Bên sở hữu dữ liệu có thể kiểm soát các quyền thuộc tính của dữ liệu cục bộ thông qua các quyền thuộc tính.

Thiết bị lưu trữ dữ liệu (tức là thuộc tính) thường được gọi là máy chủ và thiết bị nhận dữ liệu từ các thiết bị khác được gọi là máy khách. Đối với đèn thông minh và điện thoại thông minh, đèn thông minh là máy chủ và điện thoại thông minh là máy khách. Sau đây là các hoạt động phổ biến giữa máy chủ và máy khách:

* *Máy khách gửi dữ liệu đến máy chủ:* Dữ liệu được truyền bằng cách ghi dữ liệu vào máy chủ. Có hai loại thao tác ghi: một là yêu cầu ghi và hai là lệnh ghi. Sự khác biệt chính giữa hai cái này là cái trước yêu cầu phản hồi (viết phản hồi) từ ngang hàng, trong khi cái sau thì không.
* *Máy chủ gửi dữ liệu đến máy khách*: Dữ liệu cập nhật được gửi từ máy chủ đến máy khách chủ yếu dưới dạng chỉ báo hoặc thông báo của máy chủ. Tương tự như thao tác ghi, điểm khác biệt chính giữa chỉ báo và thông báo là thao tác ghi yêu cầu thiết bị kia phản hồi (xác nhận) sau khi nhận được chỉ báo dữ liệu
* *Máy khách đọc dữ liệu từ máy chủ một cách chủ động:* Máy khách nhận được giá trị của các thuộc tính tương ứng từ máy chủ thông qua thao tác đọc.

### 3.6 Tổng kết các giao thức

|  |  |  |
| --- | --- | --- |
| **So sánh** | **TCP** | **UDP** |
| Độ tin cậy | Truyền tải đáng tin cậy; hỗ trợ truyền lại, kiểm soát luồng và kiểm soát tắc nghẽn | Truyền dẫn không đáng tin cậy; không hỗ trợ truyền lại, kiểm soát luồng hoặc kiểm soát tắc nghẽn |
| Sự liên quan | Hướng kết nối, với ba lần bắt tay để thiết lập kết nối và bốn lần bắt tay để ngắt kết nối; kết nối dài | Không có kết nối; truyền dữ liệu trực tiếp; kết nối ngắn |
| Đối tượng kết nối | Kết nối một-một | Phát sóng đơn hướng một-một, phát sóng một-tất cả và phát đa hướng một-nhiều |
| Tiêu đề trên cao | ≥ 20B | 8 B |
| Tốc độ truyền | Phụ thuộc vào môi trường mạng; truyền lại xảy ra trong trường hợp mất gói, làm giảm tốc độ truyền. | Nhanh chóng, độc lập với môi trường mạng và chỉ chịu trách nhiệm truyền dữ liệu lên mạng |
| Kịch bản ứng dụng | Thích hợp cho việc truyền tải đáng tin cậy, ví dụ như truyền tập tin. | Thích hợp cho việc truyền tải thời gian thực, ví dụ như điện thoại VoIP, điện thoại video, phương tiện truyền thông trực tuyến, v.v. |

Bảng 2: So sánh TCP và UDP

|  |  |  |
| --- | --- | --- |
| **So sánh** | **HTTP** | **CoAP** |
| Lớp vận chuyển | TCP | UDP |
| Tiêu đề trên cao | Có thể chứa một lượng lớn dữ liệu tiêu đề thư với chi phí cao | Tiêu đề gói được nén nhị phân để có chi phí thấp |
| Sự tiêu thụ năng lượng | Kết nối dài, tiêu thụ điện năng cao | Kết nối ngắn, tiêu thụ điện năng thấp |
| Khám phá tài nguyên | Không được hỗ trợ | Được hỗ trợ |
| Phương thức yêu cầu | Thường được kích hoạt bởi khách hàng; không có kích hoạt hoạt động bởi máy chủ. | Cả máy khách và máy chủ đều có thể chủ động kích hoạt các yêu cầu. |
| Kịch bản ứng dụng | Thích hợp cho các thiết bị có hiệu năng tốt và bộ nhớ lớn | Thích hợp cho các thiết bị có hiệu năng kém và bộ nhớ nhỏ |

Bảng 3: So sánh HTTP và CoAP

## 4. Bảo mật

### 4.1 Transport Layer Security (TLS)

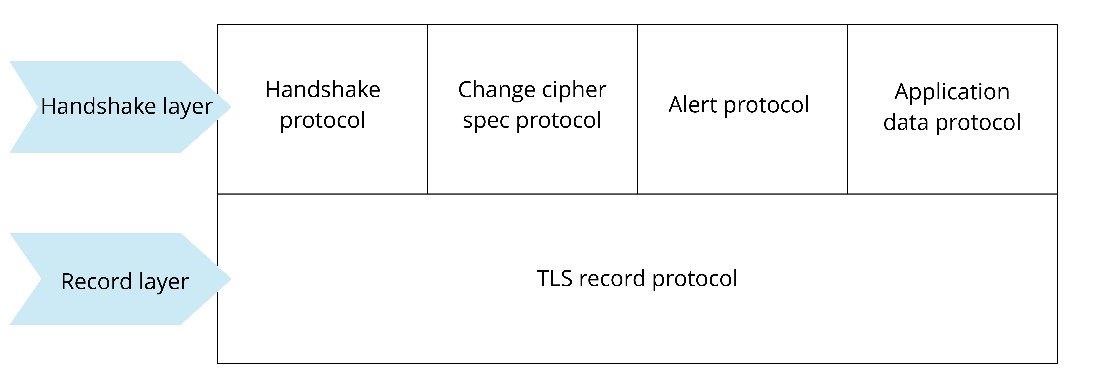
#### **4.1.1 Khái niệm**

Giao thức TLS chủ yếu giải quyết ba vấn đề mạng sau:

* *Đảm bảo tính bảo mật dữ liệu*: Tất cả dữ liệu được truyền đi được mã hóa để đảm bảo bảo vệ khỏi sự truy cập trái phép hoặc đánh cắp dữ liệu của bên thứ ba.
* *Đảm bảo tính toàn vẹn dữ liệu:* Tất cả dữ liệu đều được bảo vệ bằng cơ chế xác minh, do đó, mọi hành vi giả mạo sẽ bị cả hai bên liên quan đến giao tiếp phát hiện ngay lập tức.
* *Đảm bảo tính xác thực và xác minh danh tính của cả hai bên tham gia trao đổi dữ liệu:* Xác thực chứng chỉ có thể được cả hai bên sử dụng trong giao tiếp để đảm bảo tính hợp pháp cho danh tính của họ.

#### **4.1.2 Cách hoạt động**

Giao thức TLS có thể được chia thành hai phần. Record layer sử dụng khóa được thỏa thuận giữa máy khách và máy chủ để mã hóa và truyền dữ liệu. Handshake layer đàm phán giữa máy khách và máy chủ để xác định một tập hợp các chuỗi khóa để mã hóa truyền dữ liệu. Mô hình giao thức TLS được hiển thị trong hình dưới, trong đó Handshake layer bao gồm bốn giao thức phụ: giao thức bắt tay, giao thức thông số mật mã thay đổi, giao thức dữ liệu ứng dụng và giao thức cảnh báo.

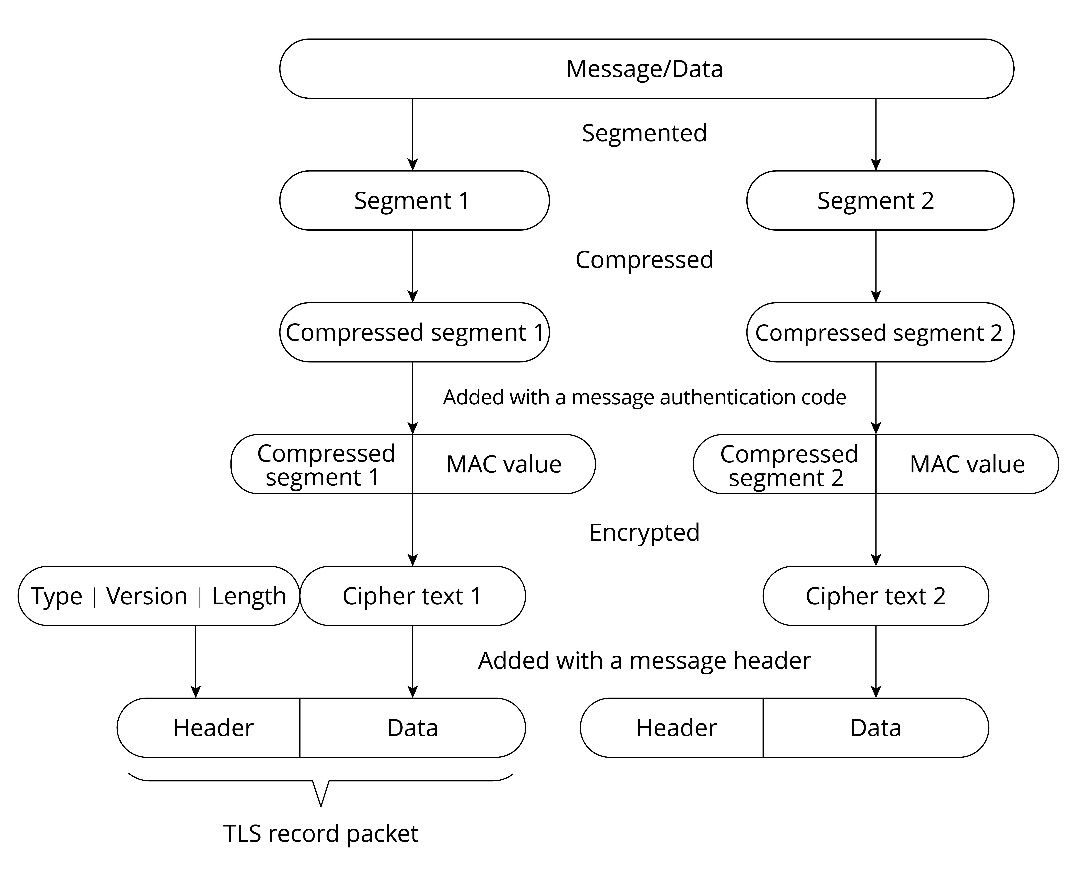


Hình 4: Mô hình giao thức TLS

#### 4.1.3 Record Layer

Record layer chịu trách nhiệm về tất cả dữ liệu cơ bản được trao đổi ở lớp vận chuyển và có thể mã hóa dữ liệu. Mỗi bản ghi TLS bắt đầu bằng một tiêu đề ngắn, bao gồm các trường Loại nội dung (hoặc giao thức con), Phiên bản giao thức và Độ dài.

Dữ liệu cơ bản được phân đoạn (hoặc hợp nhất), nén, thêm mã xác thực tin nhắn, mã hóa và sau đó được chuyển đổi thành phần dữ liệu của bản ghi TLS.

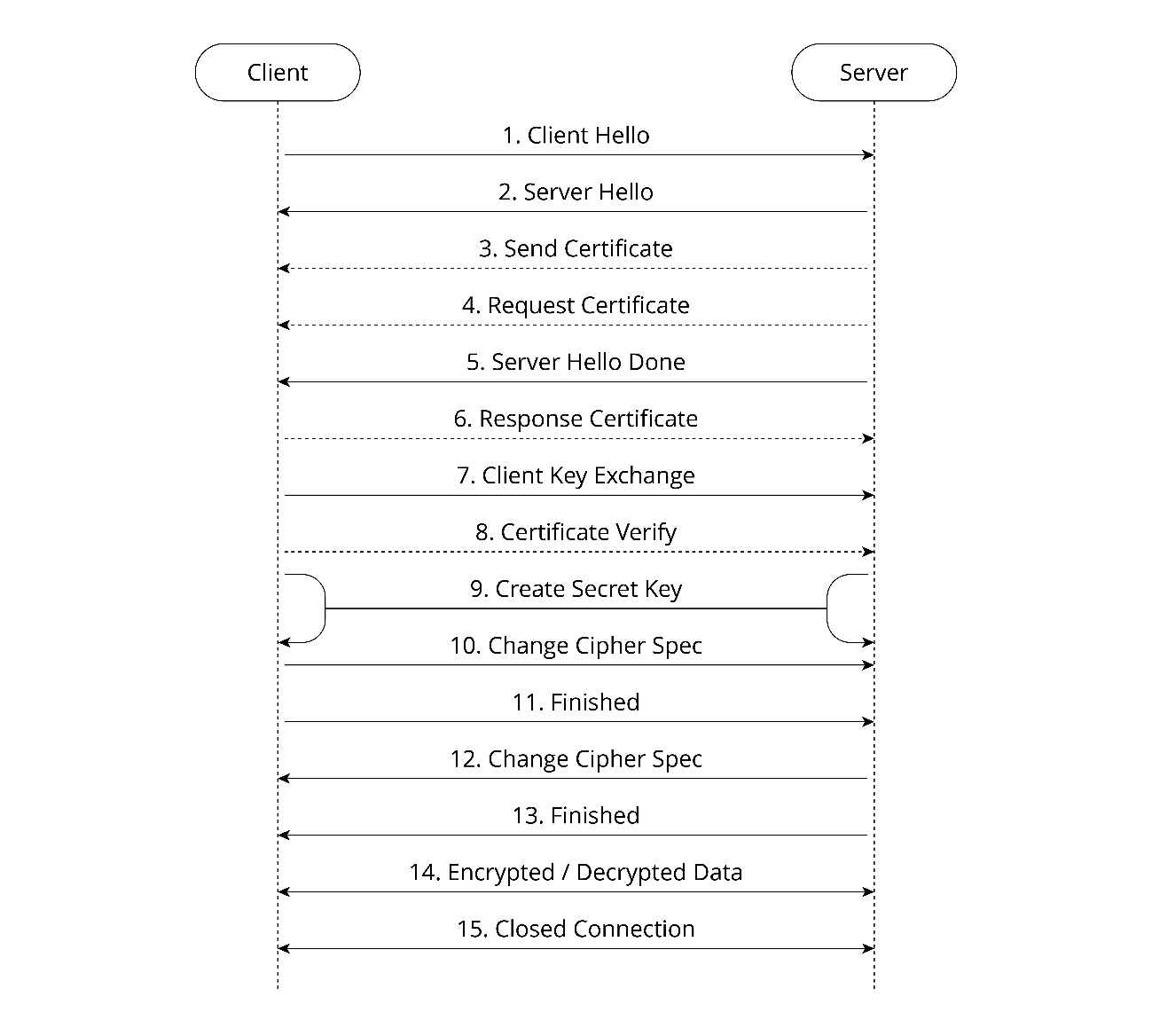


Hình 5: Cấu trúc của gói bản ghi TLS.

#### 4.1.4 Handshake layer

Gồm 4 giao thức:

* **Giao thức bắt tay**
  + Chịu trách nhiệm tạo khóa chia sẻ cần thiết cho quá trình liên lạc và thực hiện xác thực danh tính. Lưu ý rằng giao thức bắt tay không sử dụng trực tiếp các bộ mật mã. Thay vào đó, nó dựa vào mật mã khóa công khai hoặc trao đổi khóa Diffie-Hellman để thiết lập liên lạc an toàn và ngăn chặn dữ liệu bị nghe lén hoặc chặn.
* **Thay đổi giao thức thông số mật mã**
  + Chịu trách nhiệm đồng bộ hóa việc chuyển đổi mật khẩu và được sử dụng sau giao thức bắt tay. Trong quá trình bắt tay, bộ mật mã 'null', nghĩa là không mã hóa, được sử dụng. Sau khi quá trình bắt tay hoàn tất, bộ mật mã đã thương lượng sẽ được sử dụng để đảm bảo an toàn cho quá trình truyền dữ liệu tiếp theo.
* **Giao thức dữ liệu ứng dụng**
  + Được sử dụng bởi các bên giao tiếp để truyền dữ liệu. Quá trình truyền được thực hiện thông qua giao thức dữ liệu ứng dụng và giao thức ghi TLS của Handshake Layer.
* **Giao thức cảnh báo**
  + Dùng để thông báo cho bên kia khi xảy ra lỗi, chẳng hạn như ngoại lệ trong quá trình bắt tay, lỗi mã xác thực tin nhắn hoặc dữ liệu không thể giải nén được.



Hình 6: Quá trình Handshake TLS

### 4.2 Datagram Transport Layer Security (DTLS)

DTLS là giao thức dựa trên UDP phục vụ lớp ứng dụng. Giao thức TLS không thể đảm bảo tính bảo mật của dữ liệu được truyền bằng UDP. Do đó, giao thức DTLS đã được mở rộng trên kiến ​​trúc giao thức TLS hiện có để hỗ trợ UDP và trở thành một phiên bản của giao thức TLS hỗ trợ truyền gói dữ liệu. DTLS 1.0 dựa trên TLS 1.1 và DTLS 1.2 dựa trên TLS 1.2. Thuật toán mã hóa, chứng chỉ và quy trình mã hóa của giao thức DTLS về cơ bản giống với giao thức TLS, ngoại trừ những điểm khác biệt sau:

* *Ở giai đoạn bắt tay, giao thức DTLS đã bổ sung thêm cơ chế Cookie*. Giao thức DTLS đã thêm cơ chế Cookie trong phiên bản 1.0, được máy chủ sử dụng để xác minh máy khách và có thể tránh các cuộc tấn công DoS. Khi máy khách gửi tin nhắn Client Hello đến máy chủ, máy chủ không trả lời trực tiếp tin nhắn Server Hello để thực hiện quá trình bắt tay. Thay vào đó, máy chủ sẽ trả lời thông báo Yêu cầu xác minh xin chào, mang giá trị Cookie cho máy khách. Khi client nhận được tin nhắn, nó sẽ ghi giá trị Cookie vào tin nhắn Client Hello và gửi lại cho máy chủ. Sau khi nhận được, máy chủ sẽ kiểm tra danh sách Cookie cục bộ để xác định xem có cần bắt tay hay không.
* *DLTS hỗ trợ cơ chế truyền lại*. Do bản thân giao thức UDP không hỗ trợ truyền lại như giao thức TCP nên giao thức DTLS đưa ra cơ chế truyền lại. Lấy tin nhắn Client Hello ở trên làm ví dụ, sau khi client gửi tin nhắn Client Hello, client sẽ khởi động bộ đếm thời gian để nhận tin nhắn Hello Verify request do máy chủ trả lời; nếu máy chủ không trả lời trong một khoảng thời gian nhất định, máy khách sẽ gửi lại tin nhắn Client Hello. Tương tự, sau khi tin nhắn được gửi, máy chủ sẽ kích hoạt bộ hẹn giờ để theo dõi thời gian chờ và xác định xem tin nhắn có cần được gửi lại hay không.
* *DLTS hỗ trợ tiếp nhận có trật tự.* UDP không đảm bảo thứ tự các gói được gửi. Ngược lại, giao thức DTLS đã thêm một message\_seqtrường vào thông báo bắt tay. Người nhận sẽ cung cấp bộ đệm nhận để nhận các tin nhắn không đúng thứ tự (tương tự như TCP) và xử lý các tin nhắn theo thứ tự trường message\_seq.
* *DLTS hỗ trợ giới hạn kích thước gói tin*. UDP là giao thức hướng gói và TCP là giao thức hướng luồng. TCP hỗ trợ phân mảnh và tập hợp lại gói. Tuy nhiên, khi một tin nhắn UDP vượt quá đơn vị truyền tải tối đa (MTU) của lớp liên kết, nó có thể bị phân mảnh cưỡng bức ở lớp IP. Sau đó, người nhận cần xử lý gói bị phân mảnh dựa trên tiêu đề IP và tập hợp lại dữ liệu gốc. Nếu một gói bị mất, toàn bộ tin nhắn UDP sẽ không hợp lệ. Do đó, trong giao thức DTLS, các thông báo bắt tay được phân đoạn trên UDP. Điều này được thực hiện bằng cách thêm fragment\_offsettrường và fragment\_lengthtrường vào thông báo bắt tay, tương ứng với phần bù của thông báo này so với phần đầu của thông báo và độ dài của thông báo này.