**Khái niệm:**

Packets and frames là những mẩu dữ liệu nhỏ, khi hình thành cùng nhau sẽ tạo 1 thông tin lớn hơn hoặc thông báo.

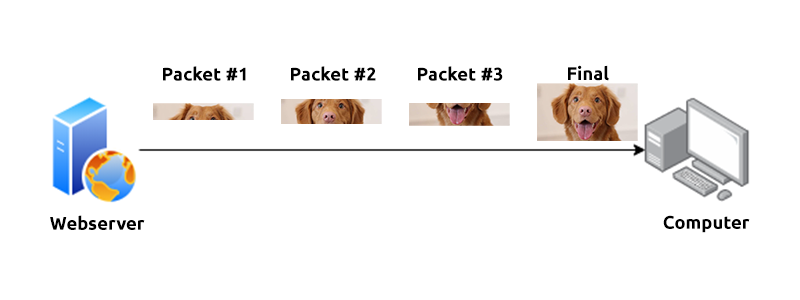
Tuy nhiên chúng là hai thứ khác nhau trong mô hình OSI. Frame là ở lớp thứ hai – liên kết dữ liệu. Điều này có nghĩa là không có địa chỉ IP.

Nghĩ nó giống như việc đặt 1 phong bì trong 1 phong bì và gửi nó đi. Phong bì đầu tiên sẽ chứa packet bạn gửi, nhưng khi nó được mở, phong bì trong vẫn còn và chứa dữ liệu (đó là frame)

Quá trình này còn gọi là sự đóng gói. Ở giai đoạn này, khi chúng ta nhắc đến địa chỉ IP thì đang nới về packets. Khi đóng gói thông tin bị loại bỏ, chúng ta đang nói về frame.

Packets là 1 cách hiệu quả để giao tiếp dữ liệu qua các thiết bị mạng khác nhau. Vì dữ liệu đã bị chia nhỏ, rất ít khi tắc nghẽn mạng hơn là những tin nhắn lớn được gửi cùng 1 lúc.

Ví dụ, khi tải 1 hình ảnh từ 1 trang web, hình ảnh không được gửi toàn bộ đến máy tính của bạn mà là những mảnh nhỏ rồi ghép lại.



Packets có nhiều cấu trúc khác nhau phụ thuộc vào loại packet đang được gửi. Như chúng ta đã thảo luận, mạng là toàn bộ những giao thức và tiêu chuẩn thiết lập 1 bộ qui tắc về cách packet được xử lí trên 1 thiết bị. Trong thực tế, có khoảng chừng 50 tỉ thiết bị kết nối cho đến cuối năm 2020, mọi thứ sẽ ngoài tầm kiểm soát nếu không có sự tiêu chuẩn.

Ví dụ về giao thức internet protocol 1 packet sử dụng giao thức này sẽ có những headers chứa các phần thông tin bổ sung cho dữ liệu đang gửi qua mạng.

|  |  |
| --- | --- |
| **Header** | **Description** |
| Time to Live | This field sets an expiry timer for the packet to not clog up your network if it never manages to reach a host or escape! |
| Checksum | This field provides integrity checking for protocols such as TCP/IP. If any data is changed, this value will be different from what was expected and therefore corrupt. |
| Source Address | The IP address of the device that the packet is being sent **from**so that data knows where to **return to**. |
| Destination Address | The device's IP address the packet is being sent to so that data knows where to travel next. |

**TCP/IP**

TCP là 1 trong những quy tắc được dùng trong mạng.

Giao thức này chứa 4 lớp và là phiên bản tóm tắt của mô hình OSI:

* Ứng dụng
* Vận chuyển
* Internet
* Giao diện mạng

Tương tự như cách hoạt động của mô hình OSI, thông tin sẽ được thêm vào ở mỗi lớp của TCP khi gói dữ liệu đi qua (packet). Quá trình này gọi là sự đóng gói.

Một đặc điểm của TCP là connection-based. Điều đó có nghĩa là TCP bắt buộc phải thiết lập 1 kết nối giữa máy khách và 1 thiết bị như máy chủ trước khi dữ liệu được gửi đến.

Do đó, TCP luôn đảm bảo bất cứ dữ liệu gửi đi sẽ được nhận ở đầu bên kia. Quá trình này gọi là three-way handshake. Một bảng so sánh những ưu điểm và khuyết điểm của TCP:

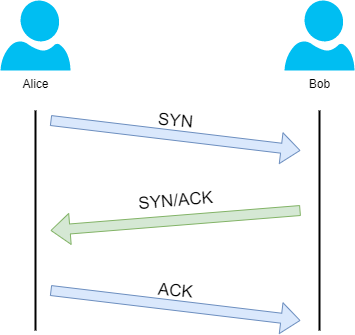
|  |  |
| --- | --- |
| **Advantages of TCP** | **Disadvantages of TCP** |
| Guarantees the integrity of data. | Requires a reliable connection between the two devices. If one small chunk of data is not received, then the entire chunk of data cannot be used and must be re-sent. |
| Capable of synchronising two devices to prevent each other from being flooded with data in the wrong order. | A slow connection can bottleneck another device as the connection will be reserved on the other device the whole time. |
| Performs a lot more processes for reliability | TCP is significantly slower than UDP because more work (computing) has to be done by the devices using this protocol. |

Những gói tin TCP chứa rất nhiều headers được thêm từ quá trình đóng gói.

|  |  |
| --- | --- |
| **Header** | **Description** |
| Source Port | This value is the port opened by the sender to send the TCP packet from. This value is chosen randomly (out of the ports from 0-65535 that aren't already in use at the time). |
| Destination Port | This value is the port number that an application or service is running on the remote host (the one receiving data); for example, a webserver running on port 80. Unlike the source port, this value is not chosen at random. |
| Source IP | This is the IP address of the device that is sending the packet. |
| Destination IP | This is the IP address of the device that the packet is destined for. |
| Sequence Number | When a connection occurs, the first piece of data transmitted is given a random number. We'll explain this more in-depth further on. |
| Acknowledgement Number | After a piece of data has been given a sequence number, the number for the next piece of data will have the sequence number + 1. We'll also explain this more in-depth further on. |
| Checksum | This value is what gives TCP integrity. A mathematical calculation is made where the output is remembered. When the receiving device performs the mathematical calculation, the data must be corrupt if the output is different from what was sent. |
| Data | This header is where the data, i.e. bytes of a file that is being transmitted, is stored. |
| Flag | This header determines how the packet should be handled by either device during the handshake process. Specific flags will determine specific behaviours, which is what we'll come on to explain below. |

TIếp theo chúng ta sẽ thảo luận về Three-way handshake thuật ngữ mô tả quá trình thiết lập kết nối giữa hai thiết bị. Three-way handshake giao tiếp bằng cách sử dụng 1 vài thông báo đặc biệt

|  |  |  |
| --- | --- | --- |
| **Step** | **Message** | **Description** |
| 1 | SYN | A SYN message is the initial packet sent by a client during the handshake. This packet is used to initiate a connection and synchronise the two devices together (we'll explain this further later on). |
| 2 | SYN/ACK | This packet is sent by the receiving device (server) to acknowledge the synchronisation attempt from the client. |
| 3 | ACK | The acknowledgement packet can be used by either the client or server to acknowledge that a series of messages/packets have been successfully received. |
| 4 | DATA | Once a connection has been established, data (such as bytes of a file) is sent via the "DATA" message. |
| 5 | FIN | This packet is used to *cleanly (properly)* close the connection after it has been complete. |
| # | RST | This packet abruptly ends all communication. This is the last resort and indicates there was some problem during the process. For example, if the service or application is not working correctly, or the system has faults such as low resources. |



Mọi dữ liệu đã gửi đều được cung cấp 1 chuỗi số ngẫu nhiên và được xây dựng lại bằng cách sử dụng chuỗi số này và tăng thêm 1.

1. SYN – Client: đây là số ngẫu nhiên ban đầu của tôi để đồng bộ hóa (0)

2. SYN/ACK – Server: đây là số ngẫu nhiên đầu tiên của tôi để đồng bộ hóa (5000) và tôi xác nhận chuỗi số của bạn là (0)

3. ACK – Client: tôi xác nhận chuỗi số của bạn là (5000), đây là một vài dữ liệu của tôi (1)

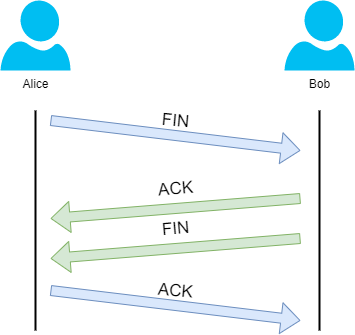
|  |  |  |
| --- | --- | --- |
| **Device** | **Initial Number Sequence (ISN)** | **Final Number Sequence** |
| Client (Sender) | 0 | 0+1=1 |
| Client (Sender) | 1 | 1+1=2 |
| Client (Sender) | 2 | 2+1=3 |

**TCP đóng kết nối:**

Đầu tiên TCP sẽ đóng kết nối khi 1 thiết bị xác định thiết bị khác đã nhận tất cả dữ liệu thành công.’

Vì TCP duy trì hệ thống những mã nguồn trên 1 thiết bị, cách tốt nhất là đóng các kết nối TCP càng sớm càng tốt.

Để bắt đầu đóng kết nối TCP, thiết bị sẽ gửi 1 gói tin FIN đến thiết bị khác. Tất nhiên với TCP, thiết bị khác cũng sẽ xác nhận gói tin đó.



Trong hình minh họa, chúng ta có thể thấy Alice đã gửi Bob 1 gói tin FIN. Bob sẽ cho Alice biết mình đã nhận được nó và anh ấy cũng muốn đóng kết nối (sử dụng FIN). Alice đã nghe thấy Bob và sẽ để Bob biết mình xác nhận điều đó.

**UDP/IP**

UDP là 1 giao thức khác được dùng để giao tiếp dữ liệu giữa các thiết bị.

Không giống như người anh em của nó TCP, UDP là 1 giao thức phi trạng thái không yêu cầu kết nối giữa hai thiết bị để truyền dữ liệu. Ví dụ three-way handshake không xảy ra, không có sự đồng hóa giữa hai thiết bị.

UDP chỉ dùng trong những trường hợp ứng dụng chịu chấp nhận mất dữ liệu (như truyền phát video hay trò chuyện bằng giọng nói). Một bảng so sánh ưu điểm và khuyết điểm của UDP như sau:

|  |  |
| --- | --- |
| **Advantages of UDP** | **Disadvantages of UDP** |
| UDP is much faster than TCP. | UDP doesn't care if the data is received or not. |
| UDP leaves the application (user software) to decide if there is any control over how quickly packets are sent. | It is quite flexible to software developers in this sense. |
| UDP does not reserve a continuous connection on a device as TCP does. | This means that unstable connections result in a terrible experience for the user. |

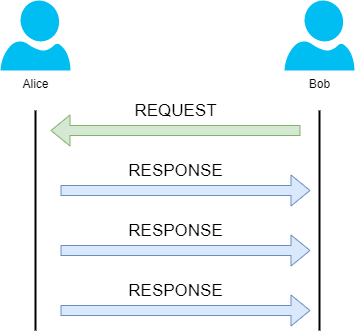
Như đã nói, không có sự kết nối giữa hai thiết bị. Điều đó có nghĩa là không quan tâm đến việc có nhận được dữ liệu hay không; không có sự toàn vẹn dữ liệu như TCP.

Những gói tin UDP thì đơn giản hơn TCP nhiều và ít headers hơn. Tuy nhiên cả hai giao thức đều có những tiêu chuẩn chung như sau:

|  |  |
| --- | --- |
| **Header** | **Description** |
| Time to Live (TTL) | This field sets an expiry timer for the packet, so it doesn't clog up your network if it never manages to reach a host or escape! |  |
| Source Address | The IP address of the device that the packet is being sent **from**, so that data knows where to **return to**. |  |
| Destination Address | The device's IP address the packet is being **sent to** so that data knows where to travel next. |  |
| Source Port | This value is the port that is opened by the sender to send the TCP packet from. This value is chosen randomly (out of the ports from 0-65535 that aren't already in use at the time). |  |
| Destination Port | This value is the port number that an application or service is running on the remote host (the one receiving data); for example, a webserver running on port 80. Unlike the source port, this value is not chosen at random. |  |
| Data | This header is where the data, i.e. bytes of a file that is being transmitted, is stored. |  |

Tiếp theo, chúng ta sẽ thảo luận về quá trình kết nối qua UDP khác với TCP như thế nào.

UDP là giao thức phi trạng thái. Không có sự xác nhận trong quá trình kết nối.



**Ports 101**

Cổng là những điểm vào cần thiết để dữ liệu có thể trao đổi với nhau.

Nghĩ về bến cảng và cổng. Những chiếc thuyền muốn cập bến cảng sẽ phải đi qua cổng: cổng phải tương thích với kích thước và cơ sở vật chất của con tàu. Khi con tàu đã sẵn sàng, nó sẽ kết nối với cổng ở bến cảng. Trong trường hợp khác, tàu du lịch không thể đi qua cổng dành cho tàu đánh cá và ngược lại.

Những cổng này qui định cái có thể đậu – nếu không tương thích, nó không thể đậu ở đây. Những thiết bị mạng cũng dùng những cổng để đưa ra những qui tắc nghiêm ngặt khi giao tiếp với thiết bị khác. Khi một kết nối được thiết lập, bất cứ dữ liệu nào được gửi hoặc nhận bởi 1 thiết bị đều sẽ đi qua những cổng này. Trong máy tính, những cổng này có giá trị từ 0 đến 65535.

Vì phạm vi cổng rất lớn, sẽ nhanh chóng mất dấu ứng dụng nào đang sử dụng dịch vụ nào. May thay chúng ta có thể gán những phần mềm ứng dụng và những hành vi với bộ qui tắc tiêu chuẩn. Ví dụ, trình duyệt web sẽ nhận được dữ liệu qua cổng 80, những nhà phát triển phần mềm có thể thiết kế trình duyệt web như Google Chrome hay Firefox để thông dịch dữ liệu.

Điều đó có nghĩa là tất cả trình duyệt web đều sử dụng chung 1 qui tắc: dữ liệu được gửi qua cổng 80.

Trong khi qui tắc cho dữ liệu web là cổng 80, một vài giao thức khác đã được gán cho 1 qui tắc tiêu chuẩn. Bất cứ cổng từ 0 đến 1024 là những cổng phổ biến. Cùng khám phá những giao thức khác dưới đây

|  |  |  |
| --- | --- | --- |
| **Protocol** | **Port Number** | **Description** |
| **F**ile **T**ransfer **P**rotocol (**FTP**) | 21 | This protocol is used by a file-sharing application built on a client-server model, meaning you can download files from a central location. |
| **S**ecure **Sh**ell (**SSH**) | 22 | This protocol is used to securely login to systems via a text-based interface for management. |
| **H**yper**T**ext Transfer Protocol (**HTTP**) | 80 | This protocol powers the World Wide Web (WWW)! Your browser uses this to download text, images and videos of web pages. |
| **H**yper**T**ext **T**ransfer **P**rotocol **S**ecure (**HTTPS**) | 443 | This protocol does the exact same as above; however, securely using encryption. |
| **S**erver **M**essage **B**lock (**SMB**) | 445 | This protocol is similar to the File Transfer Protocol (FTP); however, as well as files, SMBallows you to share devices like printers. |
| **R**emote **D**esktop **P**rotocol (**RDP**) | 3389 | This protocol is a secure means of logging in to a system using a visual desktop interface (as opposed to the text-based limitations of the SSH protocol). |

Điều đáng lưu ý là những giao thức này chỉ theo các tiêu chuẩn. Nói cách khác bạn có thể quản lí ứng dụng tương tác với những giao thức trên 1 cổng khác với cổng tiêu chuẩn (chạy 1 máy chủ web trên 8080 thay vì 80).