UDP Protocol

Alex Eagles

Contents

Table of Figures

What is UDP

User Datagram Protocol (**UDP**) is a communications protocol for time-sensitive applications like gaming, playing videos, or **Domain Name System** (**DNS**) lookups. **UDP** results in **speedier** communication because it does not spend time forming a **firm connection** with the destination before transferring the data. Because establishing the connection takes time, eliminating this step results in faster data transfer speeds.

However, **UDP** can also cause data packets to **get lost** as they go from the source to the destination. It can also make it relatively **easy** for a hacker to execute a **distributed denial-of-service** (**DDoS**) attack.

In many cases, particularly with **Transmission Control Protocol** (**TCP**), when data is transferred across the internet, it not only has to be sent from the destination but also the receiving end has to signal that it is ready for the data to arrive. Once both aspects of the communication are fulfilled, the transmission can begin. However, with **UDP**, the data is sent before a connection has been **firmly established**. This can result in problems with the data transfer, and it also presents an opportunity for hackers who seek to execute **DDoS attacks**.

How does UDP work

In comparison to other networking protocols, the process behind **UDP** is simple. A target computer is identified and the data packets, called “**datagrams**,” are sent to it. There is nothing in place to indicate the order in which the packets should arrive. There is also no process for checking if the datagrams reached the destination.

Even though **UDP** comes with **checksums**, which are meant to ensure the **integrity of the data**, and **port numbers**, which help differentiate the role the data plays at the **source** and **destination**, the lack of an obligatory handshake presents a problem. The program the user is executing with the help of **UDP** is left exposed to unreliable facets of the underlying network.

As a result, the data may get delivered, and it may not. In addition, the order in which it arrives is **not controlled**, as it is in **TCP**, so the way the data appears at the destination may be glitchy, out of order, or have blank spots.

However, in a situation where there is no need to check for errors or correct the data that has been sent, this may not pose a significant problem. This is one reason why **UDP** is used in **video applications**. Getting the video signal to its destination on time is worth the occasional glitches.

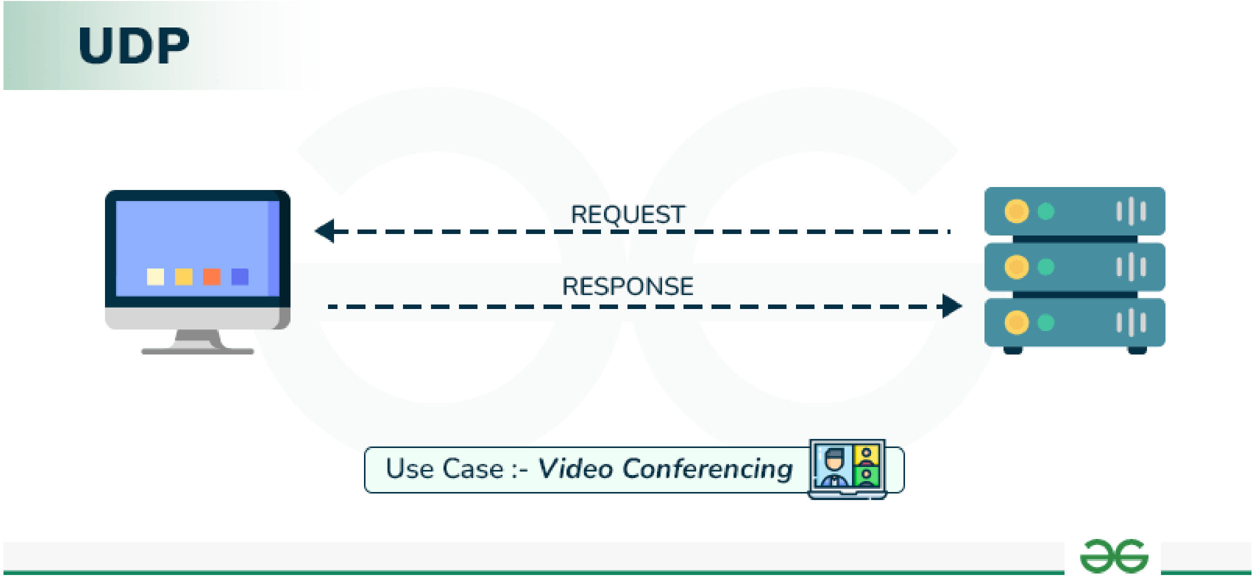

Figure 1 UDP Protocol

UDP Header

UDP header is an **8-byte** fixed and simple header, while for **TCP** it may vary from **20** bytes to **60** bytes. The first **8** Bytes contain all necessary header information, and the remaining part consists of data. **UDP** port number fields are each **16** bits long, therefore the range for port numbers is defined from **0** to **65535**; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.

- **Source Port:** Source Port is a **2** Byte long field used to identify the port number of the **source**.

- **Destination Port:** It is a **2** Byte long field, used to identify the port of the destined packet.

- **Length:** Length is the length of **UDP** including the header and the data. It is a **16-bits** field.

- **Checksum:** Checksum is **2** Bytes long field. It is the **16-bit** one's complement of the one's complement sum of the **UDP** header, the pseudo-header of information from the **IP** header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

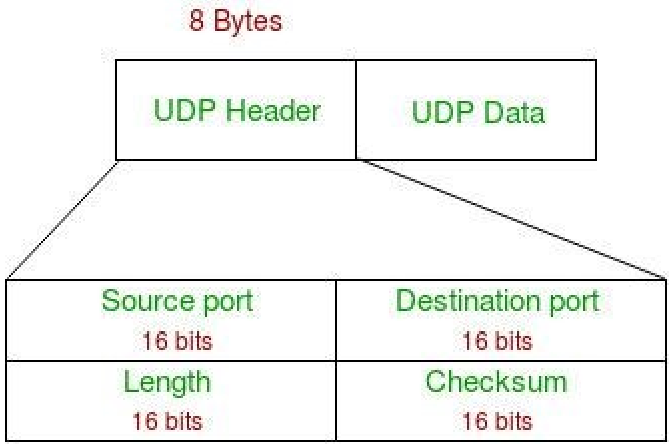

Figure 2 UDP Header

UDP Applications

- Used for simple request-response communication when the size of data is **less** and hence there is **lesser** concern about flow and error control.

- It is a suitable protocol for multicasting as **UDP** supports packet switching.

- **UDP** is used for some routing update protocols like **RIP(Routing Information Protocol).**

- Normally used for **real-time applications** which cannot tolerate uneven delays between sections of a received message.

- **VoIP (Voice over Internet Protocol)** services, such as **Skype** and **WhatsApp**, use **UDP** for real-time voice communication. The delay in voice communication can be noticeable if packets are delayed due to congestion control, so **UDP** is used to ensure fast and efficient data transmission.

- **DNS (Domain Name System)** also uses **UDP** for its query/response messages. **DNS** queries are typically small and require a **quick response time**, making **UDP** a suitable protocol for this application.

- **DHCP (Dynamic Host Configuration Protocol)** uses **UDP** to dynamically assign **IP** addresses to devices on a network. **DHCP** messages are typically small, and the delay caused by packet loss or retransmission is generally not critical for this application.

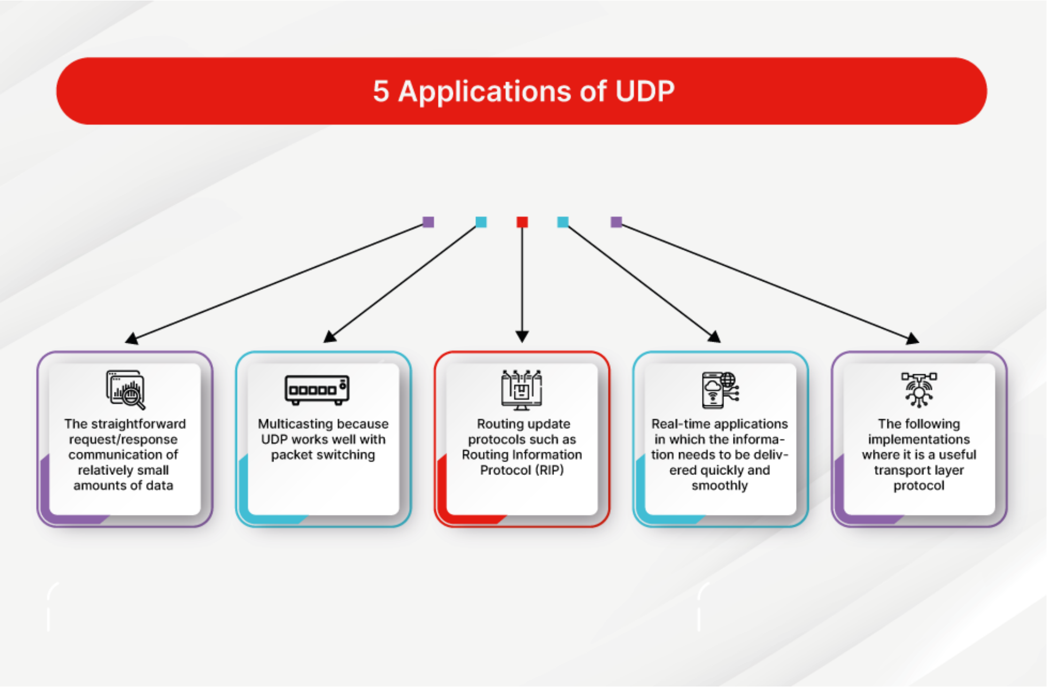
5 Applications of UDP
The straightforward request/response communication of relatively small amounts of data
Multicasting because UDP works well with packet switching
Routing update protocols such as Routing Information Protocol (RIP)
Real-time applications in which the information needs to be delivered quickly and smoothly
The following implementations where it is a useful transport layer protocol

Figure 3 UDP Applications

UDP Advantages and Disadvantages

Advantages of UDP

- **Speed: UDP is faster** than **TCP** because it does not have the overhead of establishing a connection and ensuring reliable data delivery.
- **Lower latency:** Since there is **no connection** establishment, there is lower latency and **faster response time**.
- **Simplicity: UDP** has a simpler protocol design than **TCP**, making it easier to implement and manage.
- **Broadcast support: UDP** supports broadcasting to multiple recipients, making it useful for applications such as **video streaming** and **online gaming**.
- **Smaller packet size**: **UDP** uses smaller packet sizes than **TCP**, which can reduce network congestion and improve overall network performance.
- **User Datagram Protocol (UDP)** is **more efficient** in terms of both latency and bandwidth.

Disadvantages of UDP

- **No reliability**: **UDP** does not guarantee delivery of packets or order of delivery, which can lead to **missing or duplicate data.**
- **No congestion control**: **UDP** does not have congestion control, which means that it can send packets at a rate that can cause **network congestion.**
- **Vulnerable to attacks**: **UDP** is vulnerable to **denial-of-service attacks,** where an attacker can flood a network with **UDP** packets, overwhelming the network and causing it to crash.

- **Limited use cases**: **UDP** is not suitable for applications that require reliable data delivery, **such as email or file transfers**, and is better suited for applications that can tolerate some data loss, such as **video streaming or online gaming**.