WIKIPEDIA
The Free Encyclopedia

# User Datagram Protocol

In computer networking, the **User Datagram Protocol** (**UDP**) is one of the core communication protocols of the Internet protocol suite used to send messages (transported as datagrams in packets) to other hosts on an Internet Protocol (IP) network. Within an IP network, UDP does not require prior communication to set up communication channels or data paths.

**User Datagram Protocol**

| Communication protocol | |
|---|---|
| **Abbreviation** | UDP |
| **Developer(s)** | David P. Reed |
| **Introduction** | 1980 |
| **Influenced** | QUIC |
| **OSI layer** | Transport layer (4) |
| **RFC(s)** | RFC 768 (https://datatracker.ietf.org/doc/html/rfc768) |

UDP uses a simple connectionless communication model with a minimum of protocol mechanisms. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has no handshaking dialogues and thus exposes the user's program to any unreliability of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection. If error-correction facilities are needed at the network interface level, an application may instead use Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose.

UDP is suitable for purposes where error checking and correction are either not necessary or are performed in the application; UDP avoids the overhead of such processing in the protocol stack. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for packets delayed due to retransmission, which may not be an option in a real-time system.[1]

The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768 (https://datatracker.ietf.org/doc/html/rfc768).

## Attributes

UDP is a simple message-oriented transport layer protocol that is documented in RFC 768 (https://datatracker.ietf.org/doc/html/rfc768). Although UDP provides integrity verification (via checksum) of the header and payload,[2] it provides no guarantees to the upper layer protocol for message delivery and the UDP layer retains no state of UDP messages once sent. For this reason, UDP sometimes is referred to as *Unreliable Datagram Protocol*.[3] If transmission reliability is desired, it must be implemented in the user's application.

A number of UDP's attributes make it especially suited for certain applications.

- It is *transaction-oriented*, suitable for simple query-response protocols such as the Domain Name System or the Network Time Protocol.
- It provides *datagrams*, suitable for modeling other protocols such as IP tunneling or remote procedure call and the Network File System.
- It is *simple*, suitable for bootstrapping or other purposes without a full protocol stack, such as the DHCP and Trivial File Transfer Protocol.
- It is *stateless*, suitable for very large numbers of clients, such as in streaming media applications like IPTV.
- The *lack of retransmission delays* makes it suitable for real-time applications such as Voice over IP, online games, and many protocols using Real Time Streaming Protocol.
- Because it supports multicast, it is suitable for broadcast information such as in many kinds of service discovery and shared information such as Precision Time Protocol and Routing Information Protocol.

## Ports

Applications can use datagram sockets to establish host-to-host communications. An application binds a socket to its endpoint of data transmission, which is a combination of an IP address and a port. In this way, UDP provides application multiplexing. A port is a software structure that is identified by the port number, a 16-bit integer value, allowing for port numbers between 0 and 65535. Port 0 is reserved but is a permissible source port value if the sending process does not expect messages in response.

The Internet Assigned Numbers Authority (IANA) has divided port numbers into three ranges.[4] Port numbers 0 through 1023 are used for common, well-known services. On Unix-like operating systems, using one of these ports requires superuser operating permission. Port numbers 1024 through 49151 are the registered ports used for IANA-registered services. Ports 49152 through 65535 are dynamic ports that are not officially designated for any specific service and may be used for any purpose. These may also be used as ephemeral ports, which software running on the host may use to dynamically create communications endpoints as needed.[4]

## UDP datagram structure

A UDP datagram consists of a datagram *header* followed by a *data* section (the payload data for the application). The UDP datagram header consists of 4 fields, each of which is 2 bytes (16 bits):[1]

UDP datagram header

| *Offsets* | Octet | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Octet | Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 0 | 0 | Source port | | | | | | | | | | | | | | | | Destination port | | | | | | | | | | | | | |
| 4 | 32 | Length | | | | | | | | | | | | | | | | Checksum | | | | | | | | | | | | | |

The use of the *checksum* and *source port* fields is optional in IPv4 (pink background in table). In IPv6 only the *source port* field is optional.

**Source port number**

This field identifies the sender's port, when used, and should be assumed to be the port to reply to if needed. If not used, it should be zero. If the source host is the client, the port number is likely to be an ephemeral port. If the source host is the server, the port number is likely to be a well-known port number from 0 to 1023.[4]

**Destination port number**

This field identifies the receiver's port and is required. Similar to source port number, if the client is the destination host then the port number will likely be an ephemeral port number and if the destination host is the server then the port number will likely be a well-known port number.[4]

**Length**