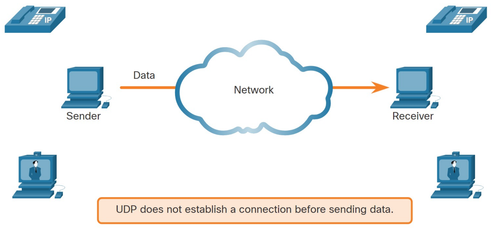
UDP Communication

14.7.1

UDP Low Overhead versus Reliability

As explained before, UDP is perfect for communications that need to be fast, like VoIP. This topic explains in detail why UDP is perfect for some types of transmissions. As shown in the figure, UDP does not establish a connection. UDP provides low overhead data transport because it has a small datagram header and no network management traffic.



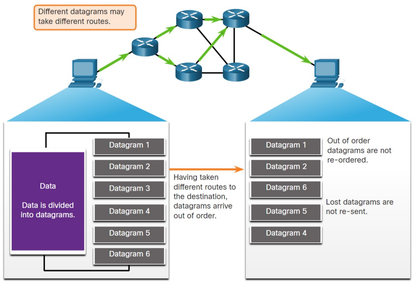
14.7.2

UDP Datagram Reassembly

Like segments with TCP, when UDP datagrams are sent to a destination, they often take different paths and arrive in the wrong order. UDP does not track sequence numbers the way TCP does. UDP has no way to reorder the datagrams into their transmission order, as shown in the figure.

Therefore, UDP simply reassembles the data in the order that it was received and forwards it to the application. If the data sequence is important to the application, the application must identify the proper sequence and determine how the data should be processed.

UDP: Connectionless and Unreliable

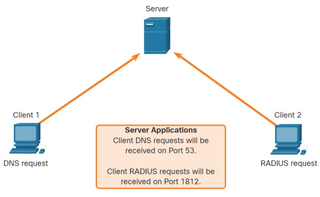


14.7.3

UDP Server Processes and Requests

Like TCP-based applications, UDP-based server applications are assigned well-known or registered port numbers, as shown in the figure. When these applications or processes are running on a server, they accept the data matched with the assigned port number. When UDP receives a datagram destined for one of these ports, it forwards the application data to the appropriate application based on its port number.

UDP Server Listening for Requests



Note: The Remote Authentication Dial-in User Service (RADIUS) server shown in the figure provides authentication, authorization, and accounting services to manage user access. The operation of RADIUS is beyond the scope for this course.

14.7.4

UDP Client Processes

As with TCP, client-server communication is initiated by a client application that requests data from a server process. The UDP client process dynamically selects a port number from the range of port numbers and uses this as the source port for the conversation. The destination port is usually the well-known or registered port number assigned to the server process.

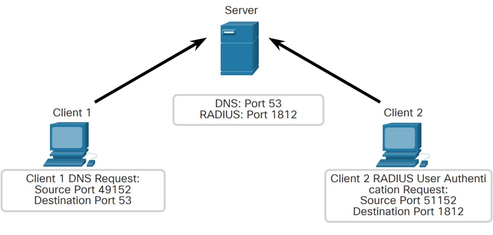
After a client has selected the source and destination ports, the same pair of ports are used in the header of all datagrams in the transaction. For the data returning to the client from the server, the source and destination port numbers in the datagram header are reversed.

Select each tab for an illustration of two hosts requesting services from the DNS and RADIUS authentication server.

Clients Sending UDP Requests | UDP Request Destination Ports | UDP Request Source Ports | UDP Response Destination | UDP Response Source Ports

Clients Sending UDP Requests

Client 1 is sending a DNS request while Client 2 is requesting RADIUS authentication services of the same server.



14.7.5

Check Your Understanding - UDP Communication

Check your understanding of the UDP communication by choosing the correct answer to the following questions.

1. Why is UDP desirable for protocols that make a simple request and reply transactions?

- Flow Control
- Low overhead
- Reliability
- Same-order delivery

2. Which UDP datagram reassembly statement is true?

- UDP does not reassemble the data.
- UDP reassembles the data in the order that it was received.
- UDP reassembles the data using control bits.
- UDP reassembles the data using sequence numbers.

3. Which of the following would be valid source and destination ports for a host connecting to a DNS server?

- Source: 53, Destination: 49152
- Source: 1812, Destination: 49152
- Source: 49152, Destination: 53
- Source: 49152, Destination: 1812

Check | Show Me | Reset