**UDP/TCP Connection or Session Multiplexing**

Connection or session multiplexing occurs at the Transport layer. It allows multiple applications to send and receive data simultaneously. It assigns a unique number to each individual session or connection to keep it separate from others. This unique number is known as port number.

This tutorial is the second part of the article "Similarities and Differences between TCP and UDP explained with functions" This tutorial explains following CCNA topic.

Compare and contrast TCP and UDP protocols

Other parts of this article are following.

[Segmentation Explained with TCP and UDP Header](https://www.computernetworkingnotes.com/ccna-study-guide/segmentation-explained-with-tcp-and-udp-header.html)

*This tutorial is the first part of the article. It explains segmentation process along with TCP/UDP header in detail.*

[TCP Features and Functions Explained with Examples](https://www.computernetworkingnotes.com/ccna-study-guide/tcp-features-and-functions-explained-with-examples.html)

*This tutorial is the last part of the article. It explains how TCP provides guaranteed data delivery through its protocol specific features.*

In each session or connection, two port numbers are used; source port and destination port. **Source port number** is used to identify the session or connection while **destination port number** is used to identify the application that processes the data at destination host.

To assign the source port number and the destination port number, both Transport layer protocols TCP and UDP use two fields in segment header; source port field and destination port field.

The port number field is 16 bits in length that allows a total of 65536 (from 0 to 65535) port numbers. Port numbers are divided in three categories; well-known, registered and dynamically assigned.

|  |  |  |
| --- | --- | --- |
| Port type | Range | Description |
| Well-known | 0 to 1023 | IANA assigns these numbers to the applications those are commonly used on the Internet such as DNS and HTTP. |
| Registered | 1024 to 49151 | IANA assigns these numbers to the applications those are proprietary applications such as Shockwave and Oracle. |
| Dynamically assigned | 49152 to 65535 | Sending device assigns these numbers to the sessions. |

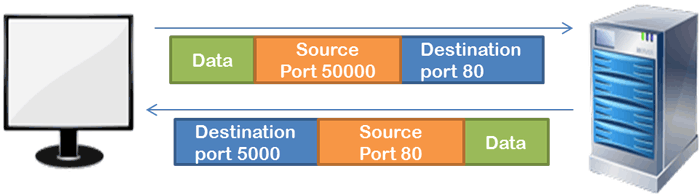
Well-known and registered port numbers are used in the destination port field while dynamically assigned port numbers are used in the source port field.

Transport layer, at source host, assigns a separate port number to each individual session from the dynamically assigned port numbers. When it initiates a new session, it picks a currently unused dynamic port number from 49152 to 65535 and assigns it to the session. All segments which are sent through this session use the assigned port number as the source port number. For destination port number, the port number of destination application is used.

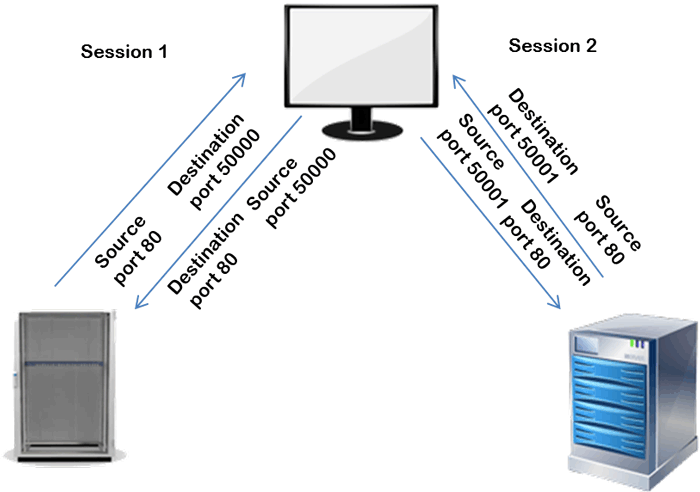
Transport layer at destination host, upon receiving segments from the source host, checks the destination port field in each segment to know by which application that segment should be processed. After processing, when the destination application returns the data, the transport layer at destination host uses the same port numbers in reverse. It uses the port number of application by which the segment was processed as source port number and the port number from which the segment was received as destination port number.

Let’s take an example. A host wants to access a website. So it initiates a TCP session. Transport layer assigns a dynamic port number 50000 to this session. Since this session is initiated to communicate with a webserver which uses HTTP protocol to serve the websites and the HTTP protocol uses the port number 80, it uses destination port number 80.

Upon receiving segments from this host, transport layer at webserver checks the destination port number in segments headers. From destination port number, it knows that the source wants to communicate with an application that uses HTTP protocol. While responding to this host, it uses the destination port number 5000 and the source port number 80.



Now suppose, host wants to access another website from same or other webserver. So it initiates a new session. Since it’s a new session, transport layer assigns a new port number to it. New port number allows it to keep its segments separate from the existing session.

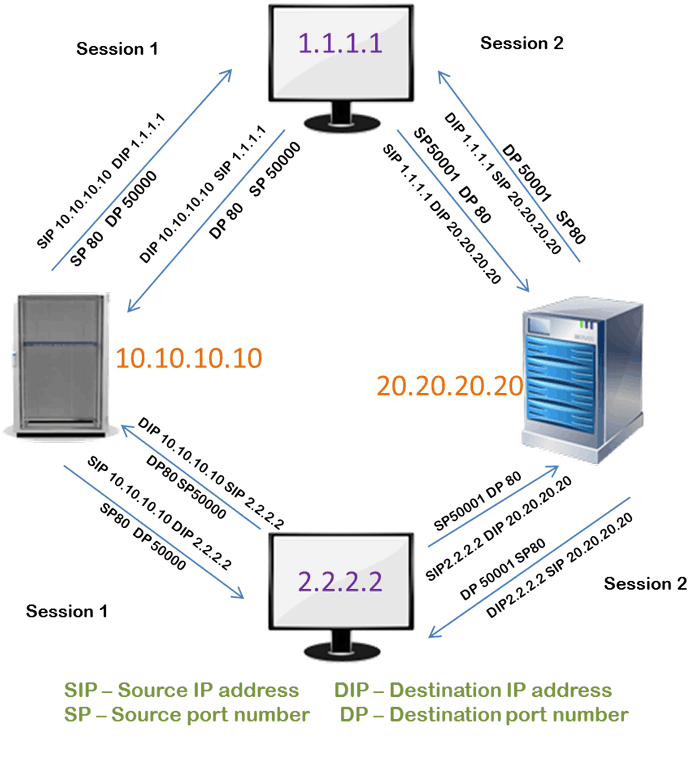


If there is only one host that access remote hosts, port numbers are sufficient to multiplex the sessions. But if there is more than one host, session multiplexing can’t be done only from the port numbers.

Let’s take an example. There are two hosts those want to access a webserver simultaneously. So they both initiate sessions. Spouse they both assign the same port number 50000 to their session. Now how will the webserver know which segment is coming from which host?

To deal with such a situation, IP address is used with the port number in multiplexing. Since both hosts have different IP addresses, destination host can easily differentiate their sessions even they both use the same source and destination port number.

Following figure shows an example of this. In this example, two hosts 1.1.1.1 and 2.2.2.2 are accessing two webservers 10.10.10.10 and 20.20.20.20 simultaneously with the same source port numbers.



In this way, to make a connection or session unique or to allow a host to connect with multiple applications simultaneously, three things are used together; Transport layer protocol, source port number and destination IP address. To refer these three things together, a technical term **socket** is used.

For example, in above figure; the host 2.2.2.2 is using the socket (10.10.10.10, TCP, 50000) to connect with the web server 10.10.10.10 while to connect with the web server 20.20.20.20, it is using the socket (20.20.20.20, TCP, 5001).

Same way, the host 1.1.1.1 is using the socket (10.10.10.10, TCP, 50000) to connect with the web server 10.10.10.10 while to connect with the web server 20.20.20.20, it is using the socket (20.20.20.20, TCP, 5001).

Both transport layer protocol TCP and UDP use socket based multiplexing to deliver the data to the correct application at the source and destination hosts