



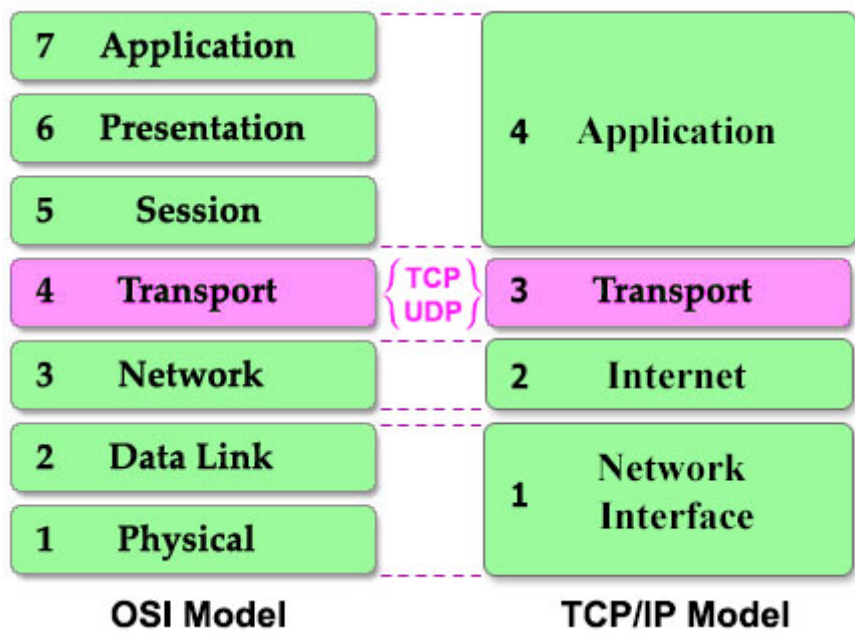
Type text to search here...

[Home](#) > TCP and UDP Tutorial

TCP and UDP Tutorial

September 20th, 2016 [Go to comments](#)

The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the two most popular protocols in the transport layer. They ensure that messages are delivered error-free, in sequence, and with no losses or duplication. The key difference between TCP and UDP is that TCP provides a wide variety of services to applications, whereas UDP does not. As a result of this, TCP is much more complex than UDP so this tutorial is dedicated to explore TCP in detail but we still compare them.



Both TCP and UDP are protocols at the Transport layer (of both OSI and TCP/IP model) but why we need both of them? The answer is:

- + TCP is slower but reliable
- + UDP is faster but unreliable

In most cases we will want to be reliable in web accessing, email communicating, file uploading... as we don't expect a few corrupted packets would destroy our whole work. With TCP, these corrupted packets will be resent or repaired to make sure everything is correct. Yes, TCP is really nice to ensure your work is accurate!

But with a price...

To guarantee the sending segments is free of error, TCP adds some bits for tracking and checking purpose so that the other end can verify and ask for missing pieces of segments. As a result of this, the segments become larger, consume more bandwidth and CPU resources to proceed.

Although UDP cannot guarantee everything is accurate like TCP but UDP is faster than TCP because it does not require additional bits for tracking and checking purpose. So which tasks need speed? Video (streaming)

and audio are ideal for this task because they are considered real-time applications. Suppose you are talking to your friend, surely you want your voice to reach your friend without any delay. It would be very weird if your friend can only hear your voice after a few seconds.

Note: Segment is the name of the data packet at Transport layer

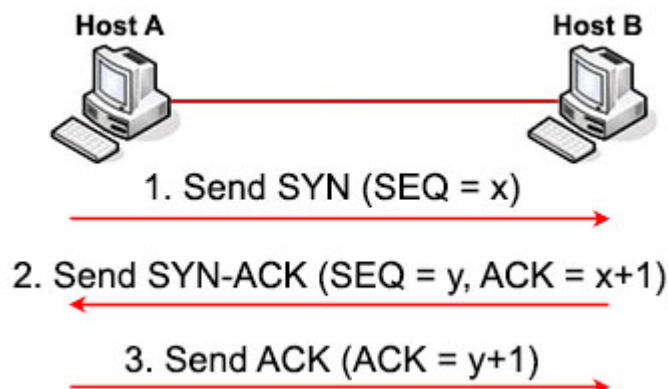
TCP can also slow down the transmission if it sees the path to the destination is too crowded. You don't want TCP to slow down your voice in traffic-jam hours either. For example when you say "Hello, how are you?", your friend at the other end may hear "Helloooooo,..... hooooooooow arrrrrrre yyyyyouu". What is an awful conversation!

Losing a few packets for voice or video is acceptable. For example if you say the word "Hello" in one second, an IP phone generates about 25 to 100 packets (just an estimation, it depends on the codec and sampling frequency) so your friend can still understand what you say even if a few packets are missing. Moreover, re-transmission the missing packets is not useful as voice and video are real-time applications and the receiving end cannot wait for the missing segments to be resent.

So now we have some basic understanding of TCP and UDP. In the next part we will learn more about TCP. Let's start with how TCP set up and terminate a connection.

TCP three-way handshake (to start the communication)

Suppose host A wants to start communicating with host B using TCP. Before they can send real data, a three-way handshake must be established first. Let's see how this process takes place:



1. First host A will send a **SYN message** (a TCP segment with SYN flag set to 1, SYN is short for SYNchronize) to indicate it wants to setup a connection with host B. This message includes a sequence (SEQ) number for tracking purpose. This sequence number can be any 32-bit number (range from 0 to 2^{32}) so we use "x" to represent it.

2. After receiving SYN message from host A, host B replies with **SYN-ACK message** (some books may call it "SYN/ACK" or "SYN, ACK" message. ACK is short for ACKnowledge). This message includes a SYN sequence number and an ACK number:

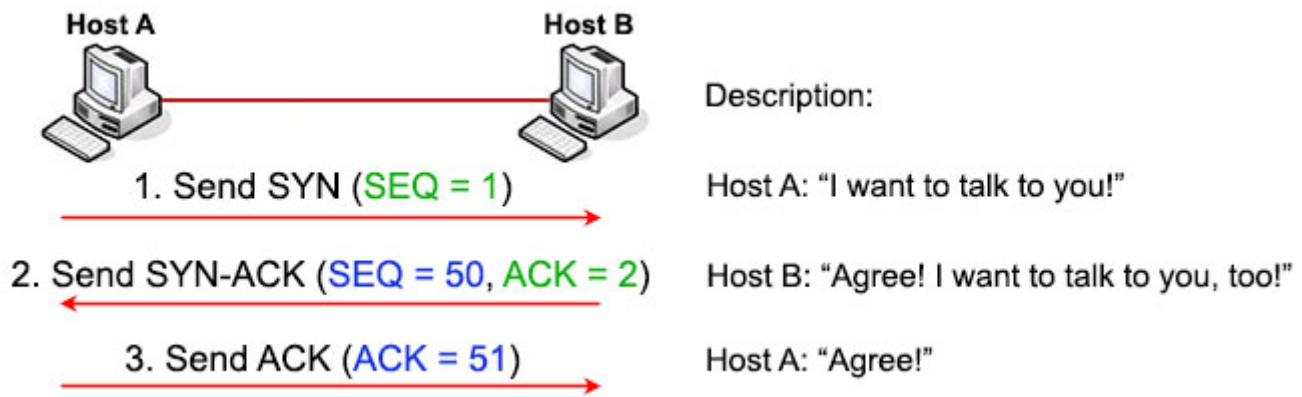
- + SYN sequence number (let's called it "y") is a random number and does not have any relationship with Host A's SYN SEQ number.

- + ACK number is the next number of Host A's SYN sequence number it received, so we represent it with "x+1". It means "I received your part. Now send me the next part (x + 1)".

The SYN-ACK message indicates host B accepts to talk to host A (via ACK part). And ask if host A still wants to talk to it as well (via SYN part).

3. After Host A received the SYN-ACK message from host B, it sends an **ACK message** with ACK number "y+1" to host B. This confirms host A still wants to talk to host B.

If you are still unclear about this process, let's assign: x = 1 and y = 50:

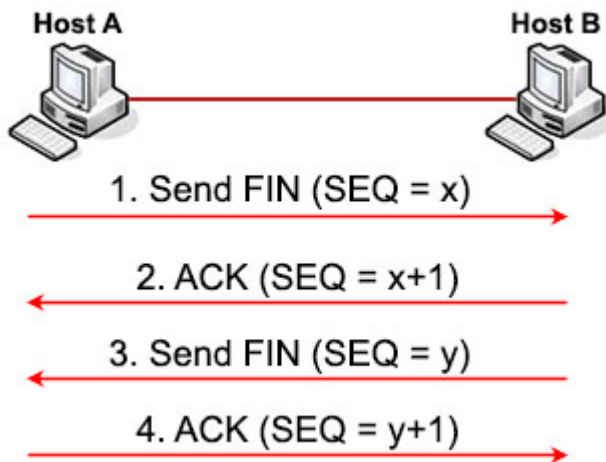


In this process, three messages need to be sent so we often call it "three-way handshake".

Nice, now you really understand TCP three-way handshake, right? Host A can start sending real traffic to host B after the three-way handshake process.

TCP also does nearly the same thing when one end wants to terminate the connection with TCP four-way termination process.

TCP four-way termination (to end the communication)



Suppose Host A wants to end the connection to host B, Host A will send a FIN message (a TCP segment with FIN flag set to 1), FIN is short for FINISH. The purpose of FIN message is to enable TCP to gracefully terminate an established connection. Host A then enters a state called the FIN-WAIT state. In FIN-WAIT state, Host A continues to receive TCP segments from Host B and proceed the segments already in the queue, but Host A will not send any additional data.

Device B will confirm it has received the FIN message with an ACK (with sequence x+1). From this point, Host B will no longer accept data from Host A. Host B can continue sending data to Host A. If Host B does not have any more data to send, it will also terminate the connection by sending a FIN message. Host A will then ACK that segment and terminate the connection.

TCP requires to establish and terminate the connection before and after exchanging real traffic so it is called **connection-oriented protocol**. UDP does not support these features so it is called **connectionless protocol**.

More formally, these terms can be defined as follows:

- + **Connection-oriented protocol**: requires a logical connection to be established between the two processes before data is exchanged
- + **Connectionless protocol**: allow data to be exchanged without setting up a link between processes

In conclusion, TCP requires the establishment (via three-way handshake) and termination (via four-way termination) of a connection. In the next part we will learn about popular TCP features.

1. Anonymous
December 9th, 2019

Can i am taking my CCNA 200-125 in 2 days... Can someone send me dumps? pilotishak @gmail(dot)com

2. Anonymous
December 16th, 2019

Kindly send CCNA latest dumps at haseebfarooq585 at gmail dot com

3. Anonymous
December 31st, 2019

Please send me the recent dump for CCNA 200-125 beckz.t.r /gmail / com
Test coming up next month.

4. Mahesh
January 8th, 2020

please send me CCNA latest dumps {email not allowed}

5. Mahesh
January 8th, 2020

mgrmlj89/gmail/com

6. Mat
January 8th, 2020

there is a very good explain about TCP/UDP protocols – I am impressed

7. Amy
January 18th, 2020

Please send dumps for CCNA 200-125 to
aspirantisee (at) gmail.com

8. Anonymous
January 28th, 2020

Can someone send me CCNA 200-125 dumps on soroyesaheedgmail.com

9. kate
January 30th, 2020

I used to prepare for my MCS D Developing Windows Azure and Web Services exam. It helped me secure 97% marks and was the only result of the 100% real exam materials I got from dumpstool.com

10. Amit
February 11th, 2020

I learned CCNA before 5 years but again I am interested to learn. Provide me the details website address or send me a pdf to learn ccna and all networking. My question?
How to be a expert in CCNA / Networking world??

11. Benard Tuju
February 13th, 2020

Taking the ccna 200-125 please send me the dump to benardtuju/yah/com

12. shiva kandel
February 18th, 2020

Please send me dump 200-105 dump, I do have an exam coming Friday. Thank you in advance. {email not allowed}. Thank you in advance

13. Anonymous
February 26th, 2020

Very helpful article

14. Anonymous
October 31st, 2021

can someone send CCNA 200-301 dumps to earvinaiken / gmail / com

Add a Comment

| | |
|----------------------|------|
| <input type="text"/> | Name |
| <div></div> | |

[Subscribe to comments feed](#)

[DHCP Sim](#) [Border Gateway Protocol](#) [BGP Tutorial](#)

Premium Member Zone

Welcome [Gurjeet singh!](#)

- [Welcome Premium Member](#)
- [CCNA – New Questions Part 5](#)
- [CCNA – New Questions Part 6](#)
- [CCNA – New Questions Part 7](#)
- [CCNA – New Questions Part 8](#)
- [CCNA – New Questions Part 9](#)
- [Composite Quizzes](#)

- [Logout](#)

CCNA 200-301

- [Basic Questions](#)
- [Topology Architecture Questions](#)

- [Cloud & Virtualization Questions](#)
- [CDP & LLDP Questions](#)
- [Switch Questions](#)
- [VLAN & Trunking Questions](#)
- [VLAN & Trunking Questions 2](#)
- [STP & VTP Questions](#)
- [EtherChannel Questions](#)
- [TCP & UDP Questions](#)
- [IP Address & Subnetting Questions](#)
- [IP Routing Questions](#)
- [IP Routing Questions 2](#)
- [OSPF Questions](#)
- [OSPF Questions 2](#)
- [EIGRP Questions](#)
- [NAT Questions](#)
- [NTP Questions](#)
- [Syslog Questions](#)
- [HSRP Questions](#)
- [Access-list Questions](#)
- [AAA Questions](#)
- [Security Questions](#)
- [Security Questions 2](#)
- [DAI Questions](#)
- [IPv6 Questions](#)
- [DNS Questions](#)
- [QoS Questions](#)
- [Port Security Questions](#)
- [Wireless Questions](#)
- [Wireless Questions 2](#)
- [SDN Questions](#)
- [DNA Center Questions](#)
- [Drag Drop Questions](#)
- [Drag Drop Questions 2](#)
- [Drag Drop Questions 3](#)
- [VPN Questions](#)
- [DHCP Questions](#)
- [Automation Questions](#)
- [Miscellaneous Questions](#)
- [CCNA FAQs & Tips](#)
- [Share your new CCNA Experience](#)

CCNA Self-Study

- [Practice CCNA GNS3 Labs](#)
- [CCNA Knowledge](#)
- [CCNA Lab Challenges](#)
- [Puppet Tutorial](#)
- [Chef Tutorial](#)
- [Ansible Tutorial](#)
- [JSON Tutorial](#)
- [Layer 2 Threats and Security Features](#)
- [AAA TACACS+ and RADIUS Tutorial](#)
- [STP Root Port Election Tutorial](#)
- [GRE Tunnel Tutorial](#)
- [Basic MPLS Tutorial](#)
- [TCP and UDP Tutorial](#)
- [Border Gateway Protocol BGP Tutorial](#)

- [Point to Point Protocol \(PPP\) Tutorial](#)
- [WAN Tutorial](#)
- [DHCP Tutorial](#)
- [Simple Network Management Protocol SNMP Tutorial](#)
- [Syslog Tutorial](#)
- [Gateway Load Balancing Protocol GLBP Tutorial](#)
- [EtherChannel Tutorial](#)
- [Hot Standby Router Protocol HSRP Tutorial](#)
- [InterVLAN Routing Tutorial](#)
- [Cisco Command Line Interface CLI](#)
- [Cisco Router Boot Sequence Tutorial](#)
- [OSI Model Tutorial](#)
- [Subnetting Tutorial – Subnetting Made Easy](#)
- [Frame Relay Tutorial](#)
- [Wireless Tutorial](#)
- [Virtual Local Area Network VLAN Tutorial](#)
- [VLAN Trunking Protocol VTP Tutorial](#)
- [IPv6 Tutorial](#)
- [Rapid Spanning Tree Protocol RSTP Tutorial](#)
- [Spanning Tree Protocol STP Tutorial](#)
- [Network Address Translation NAT Tutorial](#)
- [Access List Tutorial](#)
- [RIP Tutorial](#)
- [EIGRP Tutorial](#)
- [OSPF Tutorial](#)

Network Resources

- [Free Router Simulators](#)
 - [CCNA Website](#)
 - [ENCOR Website](#)
 - [ENSDWI Website](#)
 - [ENARSI Website](#)
 - [DevNet Website](#)
 - [CCIE R&S Website](#)
 - [Security Website](#)
 - [Wireless Website](#)
 - [Design Website](#)
 - [Data Center Website](#)
 - [Service Provider Website](#)
 - [Collaboration Website](#)

[Top](#)

