TCP and UDP Tutorial

September 20th, 2016 Go to comments

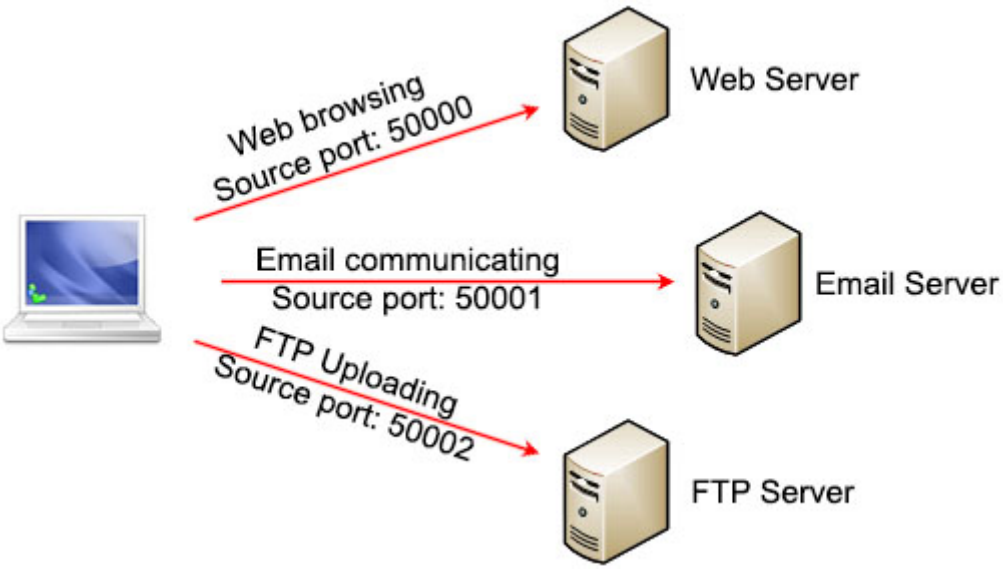
TCP Features

Some popular TCP features we will learn here are: **Multiplexing using port numbers**, **Flow control using windowing** and **Reliability** (Error Detection and Error recovery)

Multiplexing using port numbers

Suppose you are using a laptop for web browsing, email communicating and FTP uploading at the same time. All of them require using TCP while your laptop only has one IP address (with one network card) so how your laptop knows which packets received from the Internet are dedicated for which application?

Above question is solved with port numbers. Each application will use a different and available port number to communicate with outside world. For example your laptop can choose port 50000 for web browsing, port 50001 for email communicating and port 50002 for FTP uploading.



Notice that your laptop can choose any available source port but it must use pre-defined destination ports for well-known services. Port numbers are defined in three ranges:
+ Well-known port numbers (0 through 1023): assigned to key or core services that systems offer
+ Registered port numbers (1024 through 49151): assigned to industry applications and processes. For example: 1433 is assigned for Microsoft SQL Server process)
+ Dynamic port numbers (49152 through 65535): used as temporary ports for specific communications. Our laptop can use these ports for communication

The table below lists TCP ports for well-known services:

TCP Service	Description	Port
FTP	File Transfer Protocol	20, 21

SSH	Secure shell	22
Telnet	Terminal network	23
SMTP	Simple Mail Transfer Protocol	25
DNS	Domain Name Server	53
HTTP	Hyper Text Transfer Protocol	80
HTTPS	Hyper Text Transfer Protocol Secure	443

Note: There are some other well-known ports that are not listed here. The well-known ports are assigned by the Internet Assigned Numbers Authority (IANA) in the range of 0 to 1023.

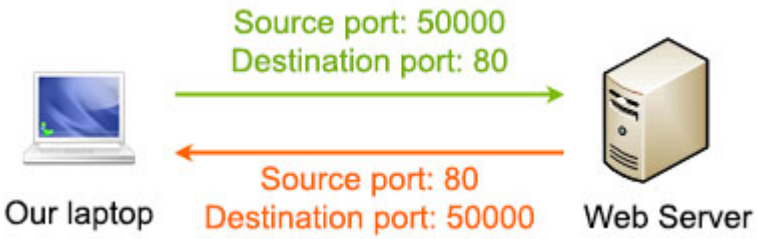
Multiplexing relies on a concept called a socket. A socket consists of three things:

+ An IP address
+ A transport protocol
+ A port number

So suppose the IP address on our laptop is 123.1.1.1 and use TCP to access web server with port 50000, we may write the socket (123.1.1.1, TCP, 50000). For web server application running on the Web Server with IP 200.1.1.1 the socket should be (200.1.1.1, TCP, 80) as the web server uses the well-known port 80 for HTTP.

The socket on each computer is unique so the connection between two sockets on two computers identify a unique connection between them. Therefore you can use multiple applications on the same computer at the same time. Each application will use a unique source port so they cannot interfere with each other.

We only mentioned about source ports but notice TCP header requires both source port and destination port. That means if our laptop wants to connect to a Web Server it must include the destination port in TCP header as well. The destination port for Web Server in this case is 80. When the Web Server replies to our laptop, it uses the laptop's source port as its destination port (50000 in this case).

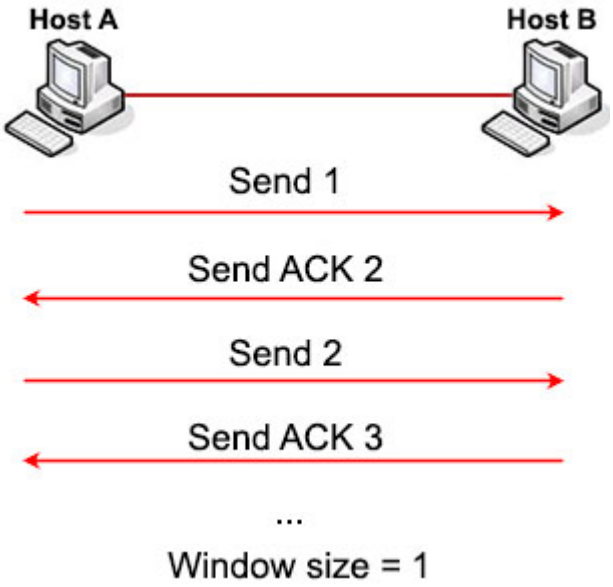


Note: Both TCP and UDP use multiplexing with port numbers for their services.

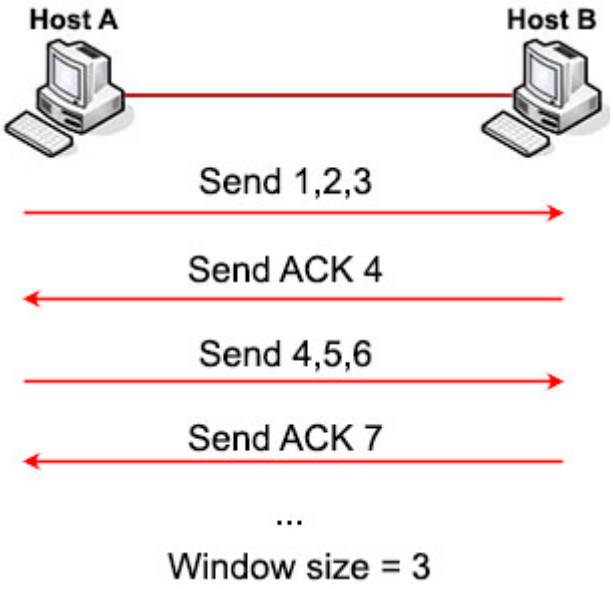
Flow-control using windowing

In the TCP header there is a field called "Window" which plays an important role in the TCP transmission. A "Window" specifies the number of segments the sender can forward without receiving an acknowledgment. It is the key to transfer data and flow control efficiently. Let's see how it works!

After the TCP connection has been established, both the client and server use this Window field to tell the other how many bytes of data it is willing to receive at one time before sending an acknowledgement to the sender. The larger the window size number (in bytes), the greater the amount of data that the host can transmit. For example, with a Window size of 1 (byte), every one byte must be acknowledged before sending the next one.



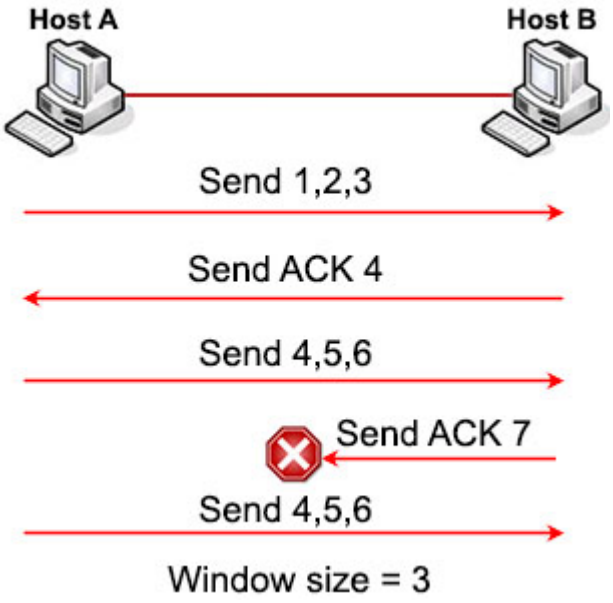
But waiting for ACK after each segment would be very inefficient. So TCP tries to increase the Window size to 3 (bytes), which means every three bytes can be received before sending the acknowledgement.



As you can see, the bigger the Window size, the fewer ACKs needed to be sent and the more efficient the transmission is. So the receiver will try to increase the Window size after each successful transmission so that the sender can send more. But the Window size cannot increase forever, TCP stops increasing the Window size when the receiver does not send an ACK (within a specific time period) or when the Window size reaches its maximum value. If a congestion occurs on the link then TCP may decrease the Window size.

The window size is variable during the lifetime of a connection so we often refer it as a “sliding window”.

If the sender does not receive the ACK in time, it knows that the segments should be resent, and that the transmission rate should be slowed down. Suppose Host A did not receive the expecting ACK 7 then it knows segments 4, 5, 6 should be resent.



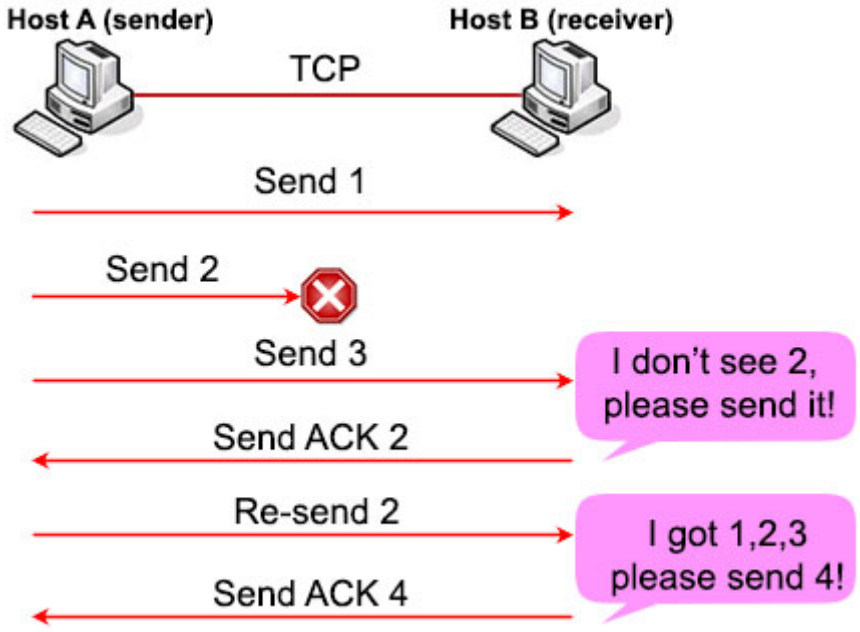
Reliability (Error Detection and Error recovery)

This is the most important feature of TCP. TCP must recover from data that is damaged, lost, duplicated during the transmission. But please grasp the difference between error detection and error recovery first:
+ **Error detection**: the detection of errors during the transmission. Error detection does not repair corrupted data, it just detects it
+ **Error recovery**: the detection of errors and repair them

To achieve error detection, TCP adds some extra bits to the data, called checksum. A TCP sender computes the checksum value based on the contents of the TCP header and data fields. This 16-bit value will be compared with the value the receiver generates using the same computation. If the values match, the receiver can believe that segment arrived intact. If the values do not match, the receiver indicates an error occurred and the segment is discarded and a notification will be sent to the receiver depending on how the TCP stack is implemented on the receiver's operating system.

To achieve error recovery, TCP uses the Sequence number (at the sender's side) and Acknowledgement fields (at the receiver's side) in the TCP header. These two fields are also used to find out lost, duplicated segments. Let's see an example.

In the transmission below, host A sends three segments 1, 2, 3 to host B. Segment 2 was lost while segment 3 arrived to Host B. Then Host B replied with an ACK 2, implying that it is expecting segment 2 next. Host A can re-send another segment 2 to recover the lost segment. If Host B receive that segment it will ask for the segment 4 (because it already has segment 3).



Error recovery

You may ask "what will happen if the ACK 2 sent from Host B is also lost?" In fact, after sending each segment Host A sets a retransmission timer, just in case the ACK is lost (or all the sending segments are lost; Host B would not send ACK in this case because it did not receive anything). If this timer expires, Host A will send all the segments again.

Note: UDP does support error detection (via checksum) but it does not support error recovery. If UDP finds a corrupted segment, it just simply drop it.

Let's sum up all things we learned about TCP and UDP so far.

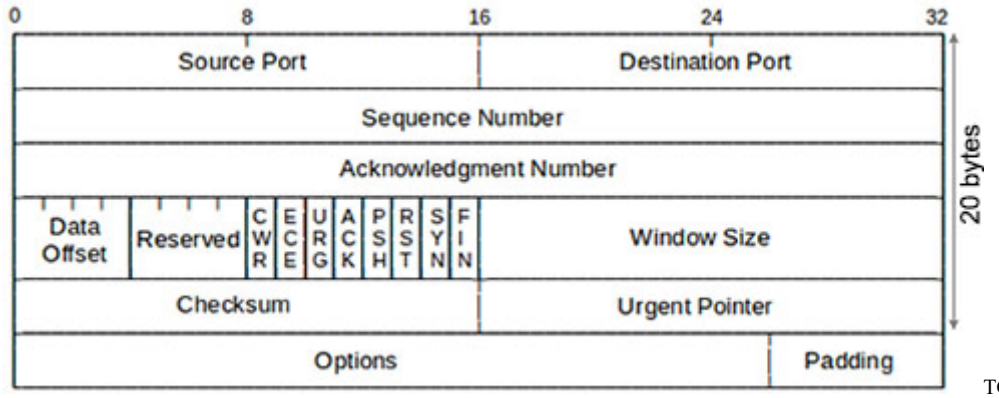
Same:

+ Both TCP and UDP operate at Transport Layer
+ Both TCP and UDP use Multiplexing via port numbers

Difference:

TCP	**UDP**
Reliable	Unreliable
Connection-oriented	Connectionless
Segment retransmission and flow control through windowing	No retransmission or windowing
Segment sequence	No sequencing
Acknowledge segment	No acknowledgement
Start and end the communication by three-way handshake and four-way termination	No action is required before and after sending real data
Support error recovery	Only support error detection

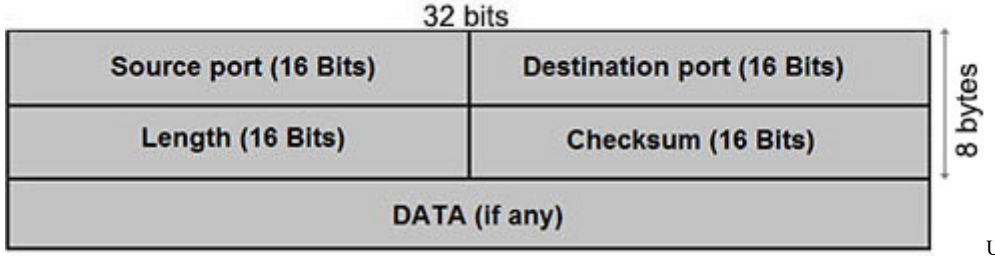
Finally we show the TCP and UDP header in detail for your reference. There are some fields which are out of scope of this tutorial.



TCP Header (20 bytes)

Notice about the FLAG fields (between the “Reserved” and “Window Size” fields). If SYN bit is turned on, it is a SYN message. If ACK bit is turned on, it is an ACK message. If both SYN and ACK bits are turned on, it is a SYN-ACK message.

And this is the UDP header:



UDP Header (8 bytes)

Pages: 1 2
Comments (14) Comments

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 MCSD 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

Name

Submit Comment

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