ICMP, TCP, UDP, IP

Protocol Types

Internet Control Message Protocol (ICMP)

On a network, whether on a Local Area Network (LAN) or a Wide Area Network (WAN), host devices will be communicating to exchange data and information between each other and sometimes an error can occur. Let's imagine you are sending a packet to a server on the internet, while your computer is initializing the connection between itself and the remote server, it provides an error stating unable to connect. As an upcoming networking professional, you may wonder why both devices are unable to successfully establish a connection amongst themselves.

Internet Control Message Protocol (ICMP) defined by **RFC 792** is typically used to provide error reporting on a network. There are many types of Internet Control Message Protocol (ICMP) messages which provide different actions and give feedback if an error occurs, and also the issue which exists.

Internet Control Message Protocol (ICMP) Message Types

There are many Internet Control Message Protocol (ICMP) message types however, we'll be discussing the main ones which will be very useful as a network professional.

ICMP Type 0 – Echo Reply

The Type 0 message is when a sender device is responding to an ICMP Type 8, Echo request.

ICMP Type 3 – Destination Unreachable

Type 3 is given then a destination cannot be found or is simply unreachable by the sender. However, ICMP Type 3 gives a bit more details by adding a Code to the message.

- Code 0 – Network Unreachable
- Code 1 – Host Unreachable
- Code 2 – Protocol Unreachable
- Code 3 – Port Unreachable

Therefore combining the ICMP Type 3 message with a unique Code gives you, the network professional a better idea to the error on the network.

ICMP Type 5 – Redirect

An ICMP Type 5 message occurs when a default gateway device such as a router notifies the sender to send the traffic directly to another gateway which exists on the same network. One reason can the second gateway device or router may have a better route to the destination or a shorter path.

ICMP Type 8 – Echo Request

The ICMP Type 8 message is used by a sender device to check for basic network connectivity between itself and the intended recipient device. Any device receiving an ICMP Type 8 message, responds with an ICMP Type 0 – Echo Reply.

ICMP Type 11 – Time Exceeded

Type 11 is given the Time to Live (TTL) expires or reaches zero (0) before reaching the intended recipient device. The last gateway which adjusts the TTL to zero (0) notified the sender using an ICMP Type 11 message as displayed below:

The **-i** parameter adjusts the Time To Live (TTL) value on the ICMP message.

```
C:\>ping 8.8.8.8 -i 4Pinging 8.8.8.8 with 32 bytes of data:
Reply from 179.60.213.149: TTL expired in transit.
Reply from 179.60.213.66: TTL expired in transit.
Reply from 179.60.213.66: TTL expired in transit.
Reply from 179.60.213.66: TTL expired in transit.

Ping statistics for 8.8.8.8:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Without adjusting the Time To Live (TTL) value of the ICMP Type 8 message, the sender received an ICMP Type 0 messages indicating successful transmission between both devices.

```
C:\>ping 8.8.8.8Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8:
bytes=32 time=52ms TTL=120
Reply from 8.8.8.8: bytes=32 time=52ms TTL=120
Reply from 8.8.8.8: bytes=32 time=52ms TTL=120
Reply from 8.8.8.8: bytes=32 time=52ms TTL=120

Ping statistics for 8.8.8.8:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round
trip times in milliseconds: Minimum = 52ms, Maximum = 52ms, Average =
52ms
```

[box type="shadow" align="" class="" width=""]Further information of Internet Control Message Protocol (ICMP) can also be found at: https://tools.ietf.org/html/rfc792 . Further information of all the Internet Control Message Protocol (ICMP) message types can be found at: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml#icmp-parameters-codes-7.[/box]

A simple and easy-to-use utility is **Ping**. The Ping utility harnesses the functionality of Internet Control Message Protocol (ICMP) and provides meaningful feedback whether communication is successful, unsuccessful, redirected, the destination host or network is unreachable, etc. The Ping utility is integrated into almost every, if not all modern day operating systems, from desktops, servers, and even mobile operating systems.

The ping command can be executed in the Windows Command Prompt or the Terminal of Linux-based Operating Systems. When a user initiates the ping command with a destination address, the ping utility would send an ICMP Type 8 message to the intended destination. The syntax for checking basic connectivity is as follows:

```
ping <ip address or hostname>
ping 8.8.8.8
ping www.google.com
```

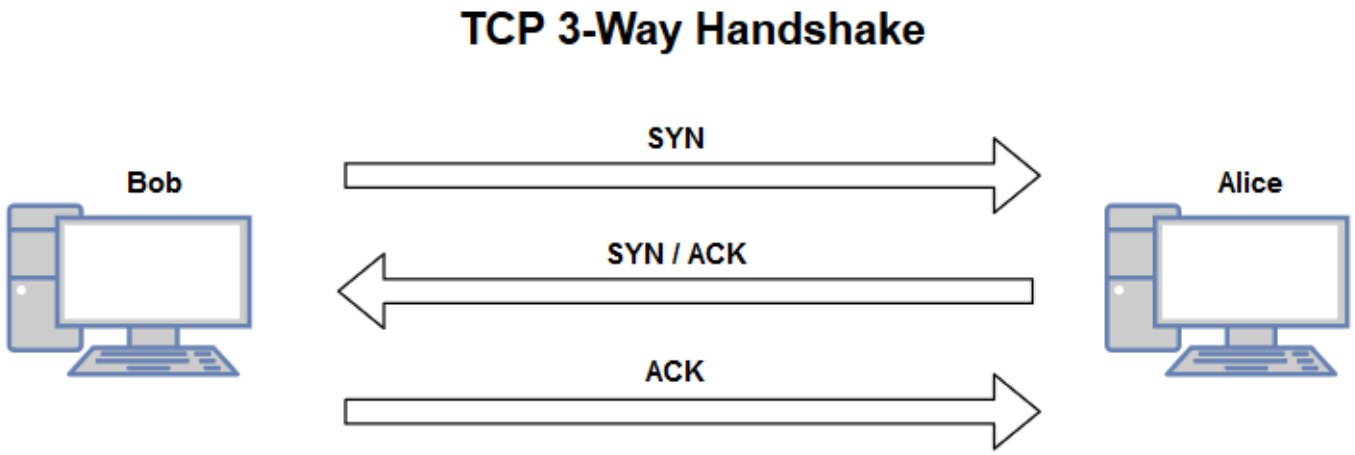
Transmission Control Protocol (TCP)

When you send a letter using your local postal service, have you ever wondered if your letter reaches the destination successfully, was your letter prioritized within the processing system of the mail service for delivery or what confirmation would you receive when the letter the is delivered successfully? Imagine in a network, these are the same concerns with devices. If one device sends a datagram to another device, whether one the same Local Area Network (LAN) or a remote network, what reassurance is given for the guarantee of the datagram (message) between sender and the receiver?

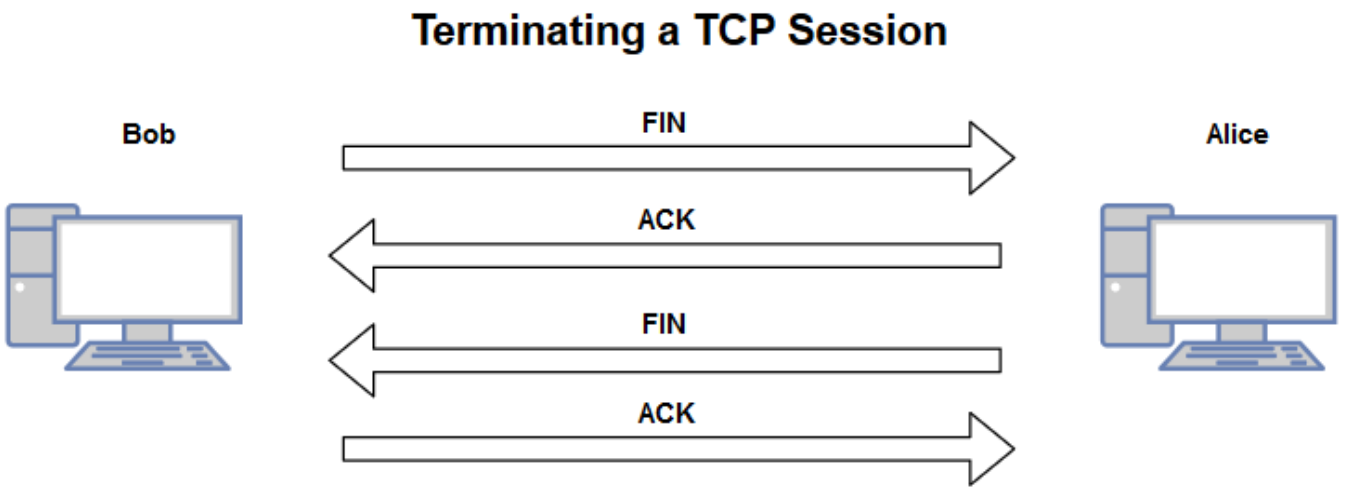
Transmission Control Protocol (TCP) defined by **RFC 793** is a **connection-oriented** protocol which operates are the Transport Layer of both the Open Systems Interconnection (OSI) reference model and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack. It is designed to provide reliable transportation of the datagrams over a network. It provides reassurance by initializing a 3-way handshake before

communicating data between the sender the receiver. Let's imagine there are two (2) devices who wants to communicate and use TCP to ensure their messages are delivered successfully. Let's use a simple analogy to further explain the TCP 3-Way Handshake, we have two (2) device, Bob and Alice. Bob wants to exchanges data with Alice but needs to ensure the data being sent are successfully delivered, so Bob decides to use the Transmission Control Protocol (TCP) to guarantee the delivery.

Bob initializes the TCP 3-Way Handshake by sending a TCP Synchronization (SYN) packet to Alice indicating he wants to establish a session or connection. Alice, upon receiving the SYN packet, responds to Bob indicating she also wants to establish a session and acknowledges receipt of the SYN packet using a TCP Synchronization and Acknowledgment (SYN/ACK) packet. Bob, upon receiving the TCP SYN packet from Alice, responds with a TCP Acknowledgement (ACK) packet. Now the TCP 3-Way Handshake is established, data can be exchanged between the two (2) devices, each datagram sent across the session between Bob and Alice, an ACK packet will be sent to confirm successful delivery of the message.



What if Bob sends a message to Alice, and Bob does not receive an ACK from Alice? In this situation, Bob would retransmit the data again after certain intervals until an ACK packet is sent back to Bob. Another question you may have is, how does Transmission Control Protocol (TCP) terminates a session gracefully? Each device sends a TCP Finish (FIN) packet to each other indicating they would like to terminate the session.



Furthermore, if we use a network protocol analyzer tools such as Wireshark, we can see the packet composition of each datagram passing across the network. The following exhibit is a capture using Wireshark during the writing of this book to demonstrate the TCP 3-Way Handshake.

Source	Destination	Protocol	Length	Info
172.16.17.12	a1488.dscd.akamai.net	TCP	62	58930 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
a1488.dscd.akamai.net	172.16.17.12	TCP	62	http(80) → 58930 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1
172.16.17.12	a1488.dscd.akamai.net	TCP	54	58930 → http(80) [ACK] Seq=1 Ack=1 Win=64240 Len=0

[box type="shadow" align="" class="" width=""]Reassemble packet in order[/box]

User Datagram Protocol (UDP)

User Datagram Protocol (UDP), defined by **RFC 768** is a connectionless protocol. This protocol also operates at the Transport Layer of both the Open Systems Interconnection (OSI) reference model and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack. However, unlike Transmission Control Protocol (TCP), the User Datagram Protocol (UDP) does not provide any guarantee or reassurance of the delivery of datagrams across a network. Not all protocols at the Application Layer uses TCP, there are many Layer 7 protocols which uses the User Datagram Protocol (UDP).

You may be wondering, why would an upper layer protocol uses UDP instead of TCP? Let do a brief recap of TCP, when devices are using TCP as their preferred Transport Layer protocol, each message sent between the sender and the receiver, an Acknowledge (ACK) packet is returned. This means if a sender such as Bob, sends one hundred (100) packets to Alice over the network, Alice would return one hundred (100) Acknowledgment (ACK) packets to Bob. Let's imagine a larger network with hundreds, thousands or even the Internet, where everyone would use TCP, the returned traffic, in this case, would the ACK packets, would create a lot of overhead in the network and therefore cause congestion. This is a bit similar to having a roadway and the number of vehicles are increasing, this would cause traffic.

Let's use another analogy, a lot of persons globally uses YouTube for many reasons. Imagine if the video traffic uses TCP instead of UDP, YouTube has millions of users daily who streams content on the site. If each user were to send a TCP ACK packet back to YouTube on that very large scale, the YouTube network and even the Internet would be congested with a lot of TCP ACK packets and would cause the network performance to degrade. Therefore, not all upper layer protocols use TCP because of this issue.

The way in which UDP behaves is simply sending datagrams without any reassurance or guarantee delivery of the message. When devices are communicating over a network, the path with each packet may take may be different from the other and therefore may be received in an out-of-order sequence. The User Datagram Protocol (UDP) does not provide any mechanisms for reassembly of the packet unlike the Transmission Control Protocol (TCP) which aids in the reassembly and reordering of the packets when they are received from the sender.

[box type="shadow" align="" class="" width=""]Voice and video traffic use UDP as the preferred Transport Layer protocol.[/box]

Comparison of TCP and UDP

Transmission Control Protocol (TCP)

- Reliable
- Uses Acknowledgments to confirm receipt of data
- Re-sends data of any of the packets are lost during transmission
- Delivers the data in sequential order and handles reassembly
- Applications: HTTP, FTP, SMTP, Telnet.

User Datagram Protocol (UDP)

- Very fast in delivery of data
- Very low overhead on the network
- Does not require any acknowledgment packets
- If packets are lost during transmission, it does not resend any lost data
- Does not send data in order or handles the reassembly
- Applications: DHCP, DNS, SNMP, TFTP, VoIP, IPTV.

[box type="shadow" align="" class="" width=""]There are protocols which uses both TCP and UDP such as DNS and SNMP.[/box]

Internet Protocol (IP)

Internet Protocol (IP) defined by RFC 791 was created for operations in interconnected systems of packet-switched computer communication networks. Internet Protocol (IP) operates at the Network Layer of the Open Systems Interconnection (OSI) reference model and the Internet Layer of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite.

However, Internet Protocol (IP) has three main characteristics:

- **Connectionless** – The sender of the message does not know if the recipient is available or not, the protocol sends the messages as is. If the message is successfully delivered to the intended recipient, the sender does not know if the message arrives or not. Since IP behaves a bit like UDP, there is not session create prior to the data communication, which leads to the receiver is not aware of any incoming messages.
- **Uses Best Effort** – Best Effort implies that Internet Protocol (IP) is unreliable. Similarly to UDP, Internet Protocol (IP) does not provide any guarantee of the data between a sender and receiver. Furthermore, if any data is lost during the transmission, IP does not have the functionality to facilitate the resending of any lost packets.
- **Media Independent** – The benefit of using Internet Protocol (IP) is, it is independent of the type of media being used for transporting the data between the sender and the receiver. At times, there are many different types of media between the sender and the receiver, such as copper cables, radio frequency, fiber optic, etc. Internet Protocol (IP) datagrams can be transported over any media type, the Data Link is responsible for formatting the Frame for each type of media as it leaves a device.