**Basic Needs for TCP/IP Communication**

Some of the applications we use require us to move data across a network from point A to point B. The Transmission Control Protocol/Internet Protocol (TCP/IP) network provides a framework for transmitting this data, and it requires some basic information from us to move this data.

[A close-up of a white background

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-five-layer-model/tcpip_basic_needs.png)

We need to specify if we want the most reliable or fastest transmissions and we need to specify where we want the data delivered. Sometimes our data is routed based on its [IP addresses](https://microchipdeveloper.com/tcpip:ip-addresses) and sometimes its routed based on its [MAC address](https://microchipdeveloper.com/tcpip:mac-addresses). The data we send needs both addressing capabilities. This information needs to be sent along with all transmitted data. We also need to physically transmit the data from one location to another.

**TCP/IP Five Layer Software Model Overview**

We need to provide this basic information needed by TCP/IP in a standard format the network can understand. This format is provided by its five-layer software model.

Each layer provides TCP/IP with the basic information it needs to move our data across the network. These layers group functions according to the task that needs to be performed. Every function in this model is targeted to help a specific layer perform its job.

[A diagram of a computer

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-five-layer-model/tcpip_5_layers.png)

Each layer only communicates with adjacent layers. Software running in a higher layer does not have to know about or perform tasks delegated to lower layer functions and vice versa. For example, the software you write for your application only needs to know how to request a connection with a remote host using the [Transport layer](https://microchipdeveloper.com/tcpip:tcp-ip-transport-layer-layer-4). It doesn’t need to know how bits are encoded before transmission. That’s the [Physical layer’s](https://microchipdeveloper.com/tcpip:tcp-ip-physical-layer-layer-1) job.

​

You are probably familiar with the [seven-layer OSI model](http://en.wikipedia.org/wiki/OSI_model). TCP/IP simplifies this model to five layers. OSI stands for Open Systems Interconnect which is a standard communication systems model. The top four layers of the seven layer OSI model have been condensed into the top two TCP/IP layers.

**TCP/IP Five**

**Layer Model Summary**

Before we discuss each layer, let’s briefly summarize what each layer does.

**Application Layer**

As you might have guessed, the [Application layer](https://microchipdeveloper.com/tcpip:common-tcp-ip-applications) is where applications requiring network communications live. Examples of these applications include email clients and web browsers. These applications use the Transport Layer to send requests to connect to remote hosts.

**Transport Layer**

The [Transport layer](https://microchipdeveloper.com/tcpip:tcp-ip-transport-layer-layer-4) establishes the connection between applications running on different hosts. It uses [TCP](https://microchipdeveloper.com/tcpip:tcp-vs-udp) for reliable connections and [UDP](https://microchipdeveloper.com/tcpip:tcp-vs-udp) for fast connections. It keeps track of the processes running in the applications above it by assigning [port](https://microchipdeveloper.com/tcpip:tcp-ip-ports) numbers to them and uses the Network layer to access the TCP/IP network.

**Network Layer**

The [Network layer](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3) is responsible for creating the packets that move across the network. It uses IP addresses to identify the packet’s source and destination.

**Data Link Layer**

The [Data Link layer](https://microchipdeveloper.com/tcpip:tcp-ip-data-link-layer-layer-2) is responsible for creating the [frames](https://microchipdeveloper.com/tcpip:tcp-ip-data-link-layer-layer-2) that move across the network. These frames encapsulate the [packets](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3) and use [MAC addresses](https://microchipdeveloper.com/tcpip:mac-addresses) to identify the source and destination.

**Physical Layer**

The [Physical layer](https://microchipdeveloper.com/tcpip:tcp-ip-physical-layer-layer-1) encodes and decodes the bits found in a frame and includes the transceiver that drives and receives the signals on the network.

[A diagram of a computer network

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-five-layer-model/tcpip_5_layer_overview.png)

**Transmit Data Using Network Layers**

Now that we know the primary job of each layer, let’s see how they work together to send and receive data across a TCP/IP network.

This is a simplified view of how the network layers work together to generate frames. Higher layers pass information to lower layers. Each layer adds information called a header to the data being passed to it. This header contains information the layer needs to perform its job. We will start at the Application layer.

**Application Layer**

The Application layer generates a message. In this case, the specific application is a web browser requesting a webpage download. This message is then sent to the Transport layer.

**Transport Layer**

The Transport layer adds the TCP or UDP header which includes the source and destination port addresses. Additional information like the packet sequence number used for TCP will also be added to the header. The data generated by the transport layer is referred to as a Segment if TCP is used, and is referred to as a Datagram if UDP is used. This segment is then sent to the Network layer.

**Network Layer**

The Network layer adds a header including the source and destination IP address to generate a packet. This packet is then sent to the Data Link layer.

**Data Link Layer**

The Data Link layer adds a header containing the MAC address information to create a frame. The frame is then sent it to the Physical layer to transmit the bits.

[![A diagram of a computer program

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-five-layer-model/transmit_data.JPG)

**TCP/IP Five Layer Software Model Terminology Reference**

[![A table with different types of data

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-five-layer-model/layer_terminology.JPG)

**TCP/IP Application Layer (Layer 5)**

The top layer or layer 5 is called the **Application** layer. This is where most [Transmission Control Protocol/Internet Protocol (TCP/IP) applications](https://microchipdeveloper.com/tcpip:common-tcp-ip-applications) live. The software you generate for your end application will typically interact with some of these applications. The most commonly used TCP/IP application is [HTTP](https://microchipdeveloper.com/tcpip:http) (Hypertext Transport Protocol) which is used for surfing the internet.

[Common TCP/IP applications >](https://microchipdeveloper.com/tcpip:common-tcp-ip-applications)

[A computer network host diagram

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-application-layer-layer-5/application_layer.png)

**TCP/IP Transport Layer (Layer 4)**

Layer 4 is the **Transport** layer. The transport layer creates virtual Transfer Control Protocol (TCP) or User Datagram Protocol (UDP) connections between network hosts.

This layer sends and receives data to and from the [applications](https://microchipdeveloper.com/tcpip:common-tcp-ip-applications) running on its host. The Transport layer assigns [port](https://microchipdeveloper.com/tcpip:tcp-ip-ports) numbers to the processes running in applications on the host and adds a TCP or UDP header to the messages received from the applications detailing the source and destination port numbers.

Note that some of the applications, specifically Telnet, SMTP, and HTTP require TCP as the transport protocol while the others use UDP.

[A computer network host diagram

Description automatically generated with medium confidence](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-transport-layer-layer-4/transport_layer.png)

Click image to enlarge.

Some [applications](https://microchipdeveloper.com/tcpip:common-tcp-ip-applications) require reliable ordered delivery of [packets](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3). The TCP protocol provides this capability. It uses error detection, retransmissions and acknowledgements. This protocol cares about your data.

Other applications don’t care if every packet is received. These applications can take advantage UDP’s lower overhead to enable faster transmissions.

Typical TCP applications include email and web browsing and typical UDP applications include VoIP and music streaming.

TCP is strictly used for point to point or unicast transmissions while UDP can also be used for multicast and broadcast transmissions.

[![A diagram of different types of computer devices

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-vs-udp/TCP_vs_UDP.JPG)

# TCP and UDP Headers

The header added to messages by the Transport layer includes more than just the source and destination [port](https://microchipdeveloper.com/tcpip:tcp-ip-ports) numbers. Here we are showing all the information included in TCP and UDP headers.

Note how the TCP protocol requires more information and overhead to guarantee data delivery.

[![A close-up of a document

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-vs-udp/TCP_UDP_headers.JPG)

**TCP/IP Network Layer (Layer 3)**

Layer 3 is the **Network** or **Internet** layer.

When transmitting data, this layer adds a header containing the source and destination [IP addresses](https://microchipdeveloper.com/tcpip:ip-addresses) to the to the data received from the [Transport layer](https://microchipdeveloper.com/tcpip:tcp-ip-transport-layer-layer-4). The packet it creates will then be forwarded to the [MAC or Data Link layer](https://microchipdeveloper.com/tcpip:tcp-ip-data-link-layer-layer-2).

When receiving data, this layer is used to determine if the packet received by the host contains the host’s IP address. If it does, the data is forwarded up to the Transport layer.

[A computer network host diagram

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-network-layer-layer-3/network_layer.png)

Click image to enlarge.

​

[Routers](https://microchipdeveloper.com/tcpip:routers) are referred to as “layer 3” devices because they route packets based on their IP addresses.

# TCP/IP IPv4 Packet Header

The Network layer header includes more than just the source and destination [IP addresses](https://microchipdeveloper.com/tcpip:ip-addresses).

[![A close-up of a document

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-network-layer-layer-3/IPv4_header.JPG)

**TCP/IP Data Link Layer (Layer 2)**

Layer 2 is the **Data Link** layer. This layer uses a **Media Access Controller (MAC)** to generate the frames that will be transmitted. As the name suggests, the MAC controls the physical transmission media. The wireless transmission media used for **Wi-Fi®** or 802.11 has different requirements from the wired transmission media used for **Ethernet** or 802.3, and therefore needs a different MAC and [PHY](https://microchipdeveloper.com/tcpip:tcp-ip-physical-layer-layer-1). Note the upper layer software is not aware of or affected by the physical interface.

[A computer network host diagram

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-data-link-layer-layer-2/data_link_layer.png)

Click image to enlarge.

When transmitting data, this layer adds a header containing the source and destination MAC addresses to the [packet](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3) received from the [Network layer](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3) (layer 3). The frame it creates will then be forwarded to the [Physical layer](https://microchipdeveloper.com/tcpip:tcp-ip-physical-layer-layer-1).

When receiving data, this layer is used to determine if the frame received by the host contains the host’s [MAC address](https://microchipdeveloper.com/tcpip:mac-addresses). If it does, the data is forwarded up to the Network layer.

Every host on the network has at least one MAC address. Laptops typically have two: one for the wired LAN and one for the wireless LAN. Home routers also typically have two MACs: one for the local network and one for the Internet.

​

Most [switches](https://microchipdeveloper.com/tcpip:switches) are referred to as “layer 2” devices because they route frames based on their MAC addresses.

# Ethernet and WiFi Frame Format

As you probably guessed, the Data Link layer adds more than just the source and destination MAC addresses to the [packet](https://microchipdeveloper.com/tcpip:tcp-ip-network-layer-layer-3). Note that the MAC for Ethernet and WiFi are different and generate different frames.

[![A screenshot of a computer

Description automatically generated]()](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-data-link-layer-layer-2/ethernet_wifi_frames.JPG)

**TCP/IP Physical Layer (Layer 1)**

Layer 1 is the **Physical** layer. It sends and receives signals on the physical wire or antenna to transmit the bits found in [frames](https://microchipdeveloper.com/tcpip:tcp-ip-data-link-layer-layer-2).

There is a PHY found at the end of every network interface (e.g. end of wire or antenna).

[A computer network host diagram

Description automatically generated](https://microchipdeveloper.com/local--files/tcpip:tcp-ip-physical-layer-layer-1/physical_layer.png)