Set 14. Internet Protocol Suite

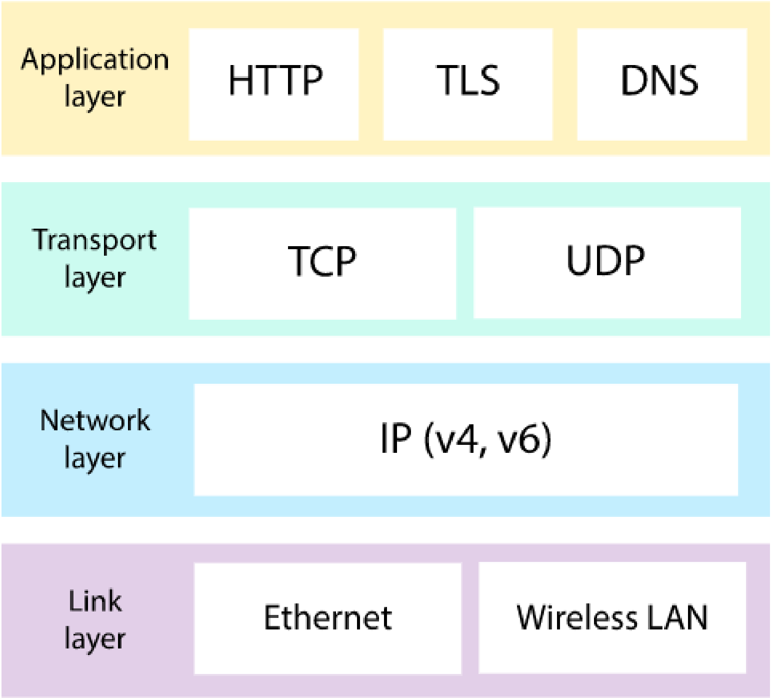
Skill 14.01: Describe the Internet protocol layers
Skill 14.02: Explain what is meant by a Protocol Stack
Skill 14.03: Explain the need for standardized protocols
Skill 14.04: Explain what is mean by “open” and why it is important
Skill 14.05: Explain the need for Open-Standard Specifications

Skill 14.01: Describe the Internet protocol layers

Skill 14.01 Concepts

As we've seen over the course of this unit, there are many protocols that power the Internet. Each protocol operates at a different layer, building functionality on top of the layer below it.

The layers of Internet protocols are often visualized in a diagram like this:

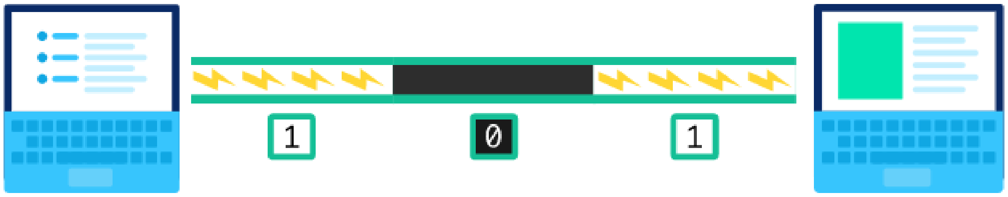


That diagram is by no means complete. There are many more protocols in the Internet protocol suite—especially at the application layer—such as SMTP for sending email and FTP for uploading files.

Let's review protocols at each layer and their contributions to the Internet.

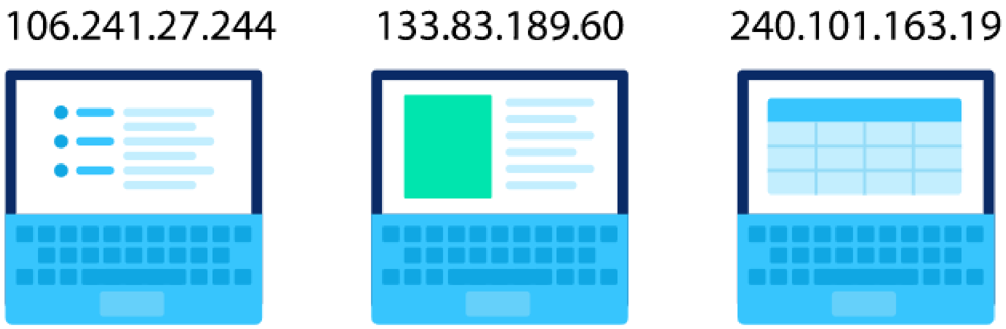
<u>Physical Layer</u>

At the bottom layer, two computing devices need a physical mechanism to send digital data to each other. They send electromagnetic signals either over a wired or wireless connection and interpret the signal as bits. The type of physical connection affects the bit rate and bandwidth.



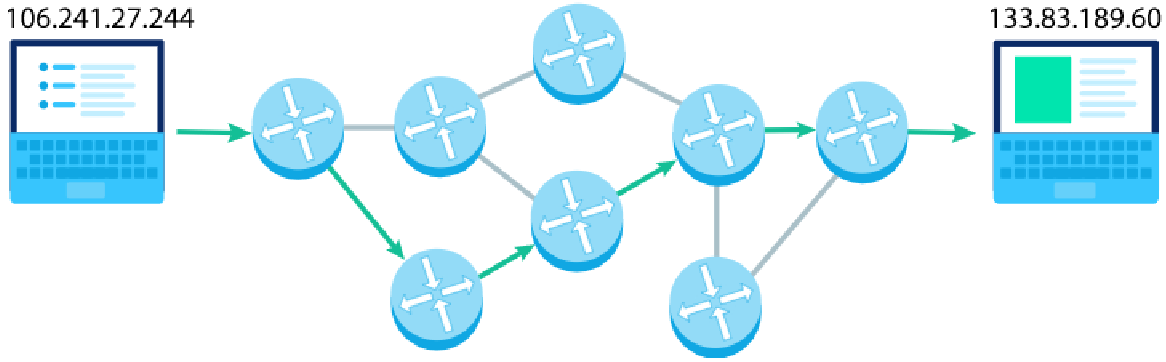
Network Layer

Once a network is bigger than two computers, we need **addressing protocols** to uniquely identify who is sending data and who should receive the data. Every node on the Internet is identified with an IP address.

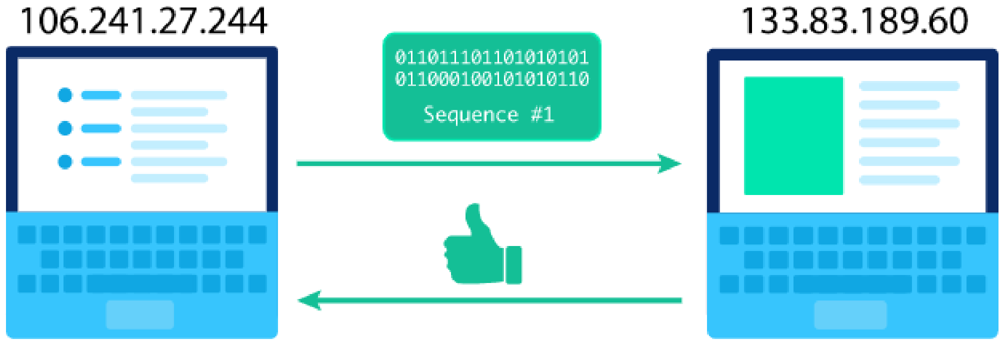


Transport Layer

The route between any two computers on the Internet isn't just a straight path from A to B. The data must pass from router to router until it finally reaches its destination, a strategy that comes from the **Internet routing protocol**.

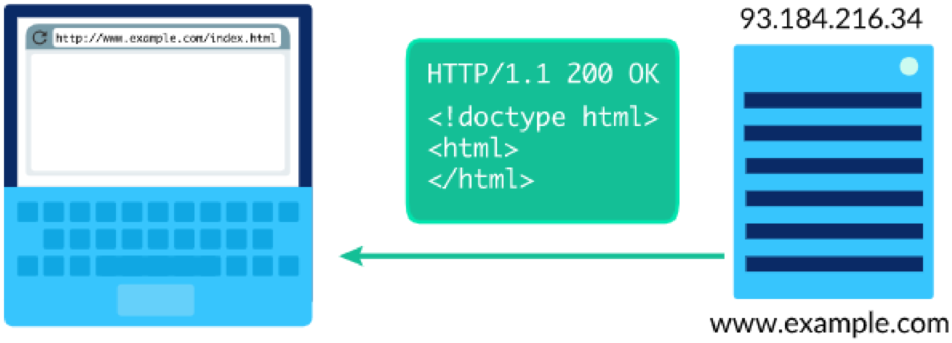


Data needs to be broken up into small packets, which are then reassembled at the destination. The **Transmission Control Protocol (TCP)** is used to ensure reliable transport of those packets, with sequencing, acknowledgement, and retries. A faster but less reliable transport protocol is the **User Datagram Protocol (UDP).**

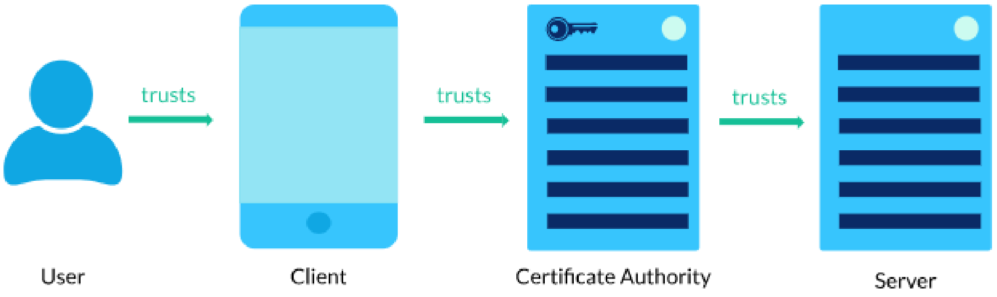


Application Layer

There are many uses for the data flowing around the Internet, such as sending emails, uploading files, or chatting online. The most common use of the Internet is the World Wide Web, with its millions of publicly viewable websites, all made possible due to the **Hyper Text Transfer Protocol (HTTP)**. We can visit a website by typing a domain name in the browser address bar, since the browser knows how to turn the domain into an IP address using the **Domain Name System (DNS)**.



When the data contains private information, it needs to be transported securely from the sender to the destination. The **Transport Layer Security (TLS) protocol** uses algorithms to encrypt the data, while certificate authorities help users trust the encryption.



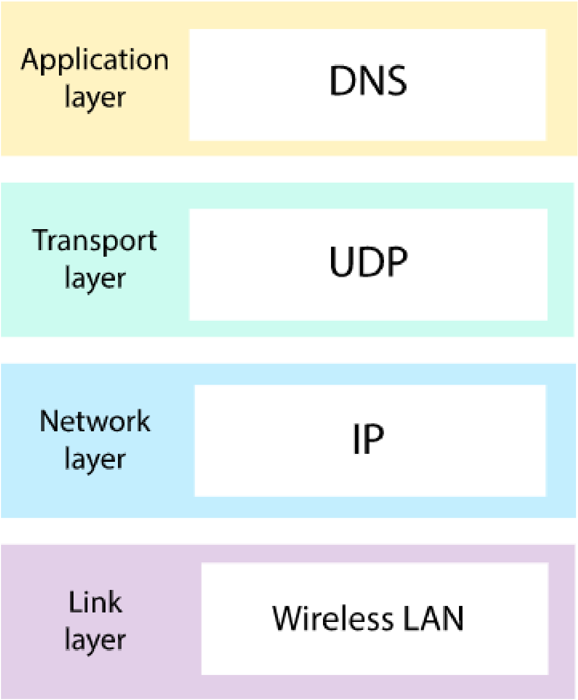
Skill 14.01 Exercises 1 & 2

Skill 14.02: Explain what is meant by a Protocol Stack

Skill 14.02 Concepts

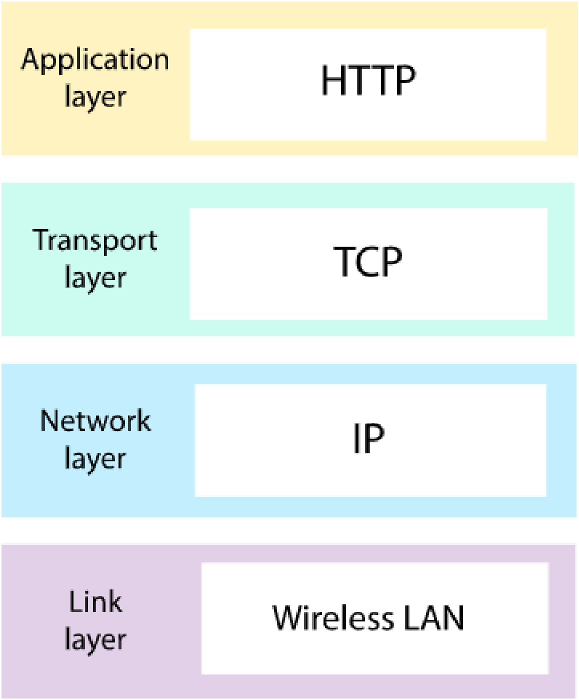
When a message is sent through the Internet, it doesn't use every protocol in the suite. It does use at least one protocol from every layer, however.

For example, when you're loading a webpage from a domain your browser has never visited before, your browser may need to make a DNS request. This stack of protocols is used when a DNS request is sent through the Internet:

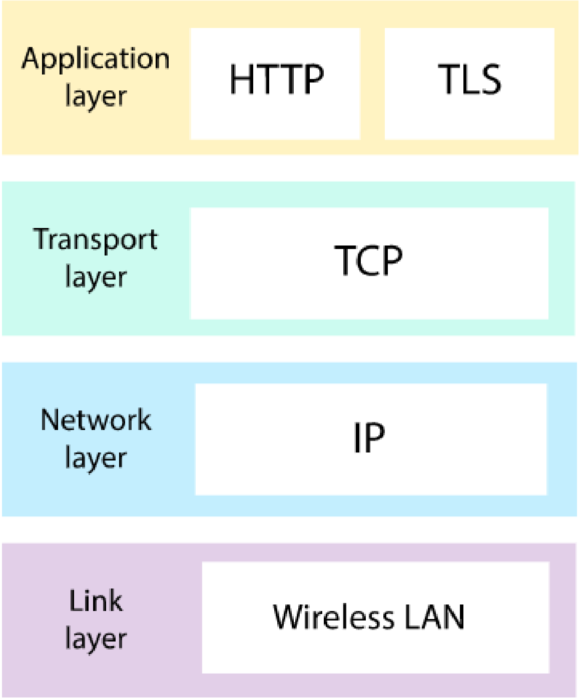


Application layer	DNS
Transport layer	UDP
Network layer	IP
Link layer	Wireless LAN

Then your browser will make an HTTP request to fetch the webpage. This protocol stack is used when an HTTP request is sent through the Internet:



If the webpage is served over HTTPS, then the stack includes multiple protocols at the application layer (both HTTP and TLS):



Skill 14.02 Exercises 1

Skill 14.03: Explain the need for standardized protocols

Skill 14.03 Concepts

Networking protocols describe how computers can communicate with each other.

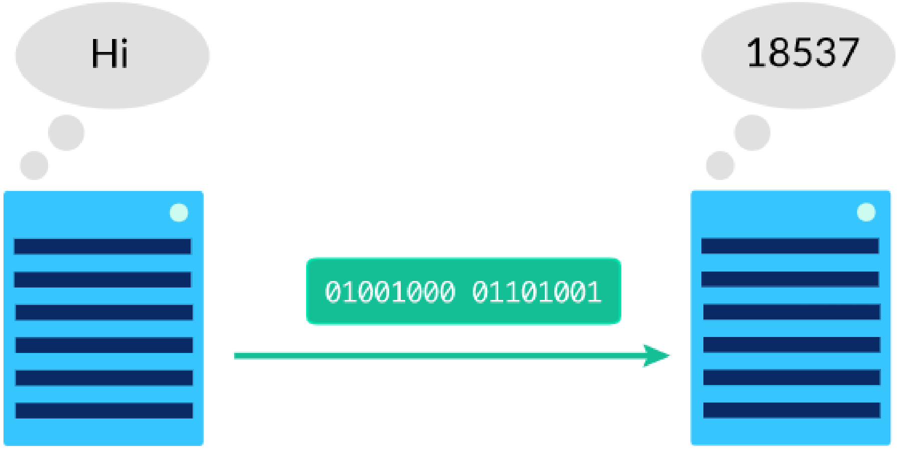
Let's imagine a world before the Internet protocols existed, a world where one computer wants to send a message to another computer.

Computer A sends this 2-byte message:

01001000 01101001

That message represents the two letters "Hi" encoded into binary using the UTF-8 standard.

Unfortunately, computer B thinks that it's receiving a number and it interprets the message as the decimal number 18537.



Each of the computers may be following a protocol about how to communicate, but they are not following the same protocol, so their attempt at communication fails. ☹

That's why standardization is so important for the technologies powering the Internet. When computers agree about the way in which they'll send messages to each other, they can understand the messages they receive.

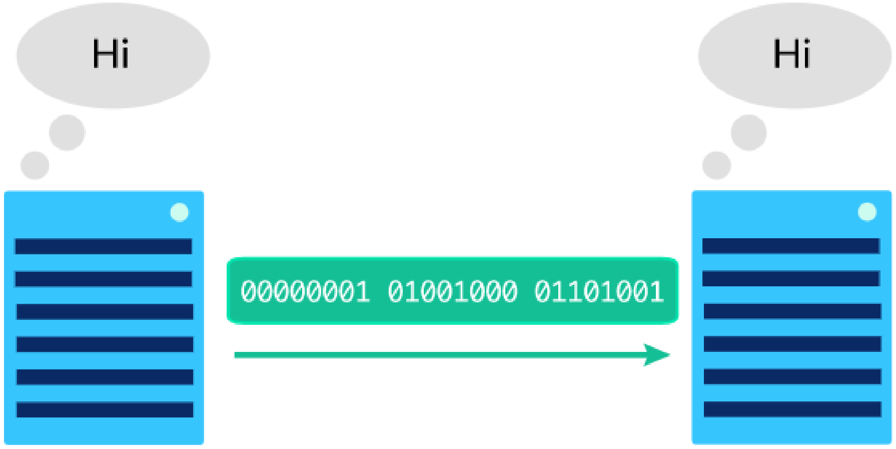
For example, the two computers could agree to follow a simple protocol where first byte indicates the type of data that's being sent. A first byte of 00000000 indicates a number and a first byte of 00000001 indicates a string of letters.

Computer A can now send this message:

00000001 01001000 01101001

Computer B receives the message, checks the first byte to discover what type of data is in the message, and then successfully decodes the rest of the message.

The computers successfully communicate! ☺



Once this protocol is written up in a document and other network administrators agree that it is a sensible protocol, that protocol is considered a standard. Any two computers could follow that standard to communicate with each other, not just the original two computers.

The Internet protocols are quite a bit more complicated than that simple protocol, of course, as there are many details to document about the complex process of communication across different types of devices.

Skill 14.04: Explain what is mean by "open" and why it is important

Skill 14.04 Concepts

An **open (nonproprietary)** protocol is one that is *not* owned by any particular company and *not* limited to a particular company's products.

The protocols in the Internet stack are open so that any computing device can follow the protocol to join the global network. You don't need to ask for permission to follow the protocols, you don't have to buy particular hardware to, and you don't need to pay money to the company that created the protocol. You simply need to read the standard and make sure the computing device follows that standard correctly.

By contrast, some communication protocols are proprietary. As an example, Skype is a voice calling application which uses its own proprietary protocol to send and receive voice calls. The Skype protocol is not publicly described anywhere, and the Skype application is closed-source. That makes it very hard for anyone besides Microsoft (the owner of Skype) to develop applications that can send or receive Skype calls.

The creators of the Internet made the protocols open on purpose. Vint Cerf, co-author of the TCP/IP protocols, said this about their development:

> *"The theory we had is that if we just specify what the protocols would look like and what software you needed to write, anybody who wanted to build a piece of internet would do that and find somebody who would be willing to connect to them. Then the system would grow organically because it didn't have any central control. And that's exactly what happened."*

Skill 14.04 Exercises 1 & 2

Skill 14.05: Explain the need for Open-Standard Specifications

Skill 14.05 Concepts

For every protocol that is both standardized and open, there is publicly viewable document describing the protocol (often called a "specification").

🔍 Visit this 88-page-long HTTP specification and skim a few sections.

Specifications are not written for the average web user. They're not even written for the average software developer. They are written for engineers that are developing new applications built on top of the protocol and need a deep understanding of how things should work.

Specifications can change over time. Sometimes there are minor errors that need to be fixed, but other times, major updates are needed to further clarify a behavior or describe a new feature. The official TCP specification for example was written in 1981 and has since accumulated 39 corrections and 4 multi-page specifications clarifying the protocol.

These days, the specifications for the Internet protocols are maintained by the **Internet Engineering Task Force (IETF)**. The IETF is an open standards committee that comes up with refinements to existing protocols and proposes entirely new protocols. The committee is made up of representatives across the Internet industry and is open to participation by anyone who has opinions about how the Internet should work.

The languages of the World Wide Web are also open standards with online specifications. The HTML living standard is maintained by the WhatWG community. CSS has many specifications and is maintained by the W3C. JavaScript is based on the ECMAScript standard.

The Internet standards communities aim to create standards that reflect the diverse needs of the applications running on top of the Internet and document those standards in clear specifications that allow any computing device to communicate and collaborate online.

Skill 14.05 Exercises 1 thru 3